

Computational Methods in Applied Sciences

Esther Andrés-Pérez · Leo M. González
Jacques Periaux · Nicolas Gauger
Domenico Quagliarella
Kyriakos Giannakoglou *Editors*

Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems



 Springer

Computational Methods in Applied Sciences

Volume 49

Series editor

Eugenio Oñate
Universitat Politècnica de Catalunya
Barcelona, Spain
onate@cimne.upc.edu

This series publishes monographs and carefully edited books inspired by the thematic conferences of ECCOMAS, the European Committee on Computational Methods in Applied Sciences. As a consequence, these volumes cover the fields of Mathematical and Computational Methods and Modelling and their applications to major areas such as Fluid Dynamics, Structural Mechanics, Semiconductor Modelling, Electromagnetics and CAD/CAM. Multidisciplinary applications of these fields to critical societal and technological problems encountered in sectors like Aerospace, Car and Ship Industry, Electronics, Energy, Finance, Chemistry, Medicine, Biosciences, Environmental sciences are of particular interest. The intent is to exchange information and to promote the transfer between the research community and industry consistent with the development and applications of computational methods in science and technology.

More information about this series at <http://www.springer.com/series/6899>

Esther Andrés-Pérez · Leo M. González
Jacques Periaux · Nicolas Gauger
Domenico Quagliarella · Kyriakos Giannakoglou
Editors

Evolutionary
and Deterministic Methods
for Design Optimization
and Control With
Applications to Industrial
and Societal Problems

 Springer

Editors

Esther Andrés-Pérez
ISDEFE
Madrid
Spain

Nicolas Gauger
Chair for Scientific Computing
Technische Universität Kaiserslautern
Kaiserslautern, Rheinland-Pfalz
Germany

Leo M. González
Fluid Mechanics and Propulsion
Department
Universidad Politécnica de Madrid
Madrid
Spain

Domenico Quagliarella
Department of Fluid Mechanics
Italian Aerospace Research Center
Capua
Italy

Jacques Periaux
International Center for Numerical
Methods in Engineering (CIMNE)
Barcelona
Spain

Kyriakos Giannakoglou
School of Mechanical Engineering
National Technical University of Athens
Athens
Greece

ISSN 1871-3033

Computational Methods in Applied Sciences

ISBN 978-3-319-89889-6

ISBN 978-3-319-89890-2 (eBook)

<https://doi.org/10.1007/978-3-319-89890-2>

Library of Congress Control Number: 2018938650

© Springer International Publishing AG, part of Springer Nature 2019

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The 12th edition of the International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN 2017) was held from 13th to 15th of September 2017 at the School of Naval Engineering, jointly organized by the Spanish National Institute for Aerospace Technology (INTA), the Technical University of Madrid (UPM) and ISDEFE in association with ECCOMAS and ERCOFTAC. A detailed information of the event, belonging to the series of ECCOMAS Thematic Conferences can be found on the website <http://eurogen2017.etsiae.upm.es/>.

This event gathered experts from Universities, Research Institutions and Industries developing or applying evolutionary and deterministic methods in design optimization with emphasis on industrial and societal applications.

EUROGEN 2017 focused particularly on:

- Metaheuristics and Evolutionary Algorithms (including Evolutionary Programming, Evolution Strategies, Genetic Algorithms, Memetic Algorithms, Artificial Immune Systems, etc.)
- Multi-objective Evolutionary Algorithms and Constraint Handling Techniques
- Adjoint-Based and One-Shot Methods
- Hybrid Optimization Methods (Gradient-Based Methods, Combinatorial Optimization Methods, etc.)
- High-Performance Computing and GPUs-Based Optimization Algorithms
- Goal-Oriented Optimization for Mesh and Meshless Methods
- Game Strategies
- Surrogate Models for Optimization
- Parallel and Distributed Evolutionary Algorithms (from LANs to GRID)
- Multi-disciplinary Optimization Methods
- Design Optimization Under Uncertainties
- Multi-criteria Decision-Making
- Topology Optimization

Some statistics of EUROGEN 2017 Conference:

- 121 total attendants (99 registered) from 16 different countries around the world, including:
 - Six (6) plenary invited speakers:
 - Prof. Juan Jose Alonso (Stanford University, USA): *‘Supersonic Low-Boom Aircraft Design Using the Open-Source SU2 Framework’*.
 - Prof. Sancho Salcedo (University of Alcala, Spain) *‘Machine Learning algorithms for prediction problems in energy applications’*.
 - Dr. Adel Abbas (Ex-Airbus Head of Aerodynamics research and technology, Spain) *‘Aircraft Design Optimization An integrated and multidisciplinary design chain’*.
 - Prof. Joaquim Martins (Univ. Michigan, USA) *‘Practical wing design via numerical optimization—Are we there yet?’*.
 - Prof. Shigeru Obayashi (Tohoku University, Japan) *‘Multi-Objective Design Exploration - Fusion of Optimization and Data Mining’*.
 - Prof. Johan Meyers (KU Leuven, Belgium) *‘Adjoint-based optimization of wind-farm control in large-eddy simulations’*.
 - Ten (10) Mini Symposia:
 - *‘Multi-disciplinary design optimization’*. Organized by A. Riccardi, E. Minisci and M. Vasile (University of Strathclyde).
 - *‘Surrogate-assisted Optimization of Real World problems’*. Organized by D. González (AIRBUS) and E. Iuliano (CIRA).
 - *‘Adjoint Methods for Optimisation, Mesh Adaptation and Uncertainty Quantification’*. Organized by J. Mueller (Queen Mary University), K. Giannakoglou (NTUA), T. Verstraete (VKI).
 - *‘Extension of fixed point PDE solvers for optimal design - Methods and Applications’*. Organized by N. Gauger and L. Kusch (TU Kaiserslautern).
 - *‘Optimum design applications in structural and civil engineering’*. Organized by D. Greiner (ULPGC), J. Magalhaes-Mendes (Politécnico do Porto) and J. Periaux (CIMNE).
 - *‘Sensitivity and adjoint methods for optimization in flow stability problems’*. Organized by E. Valero, A. Martinez-Cava and A. Rueda (UPM).
 - *‘Optimization under uncertainty’*. Organized by D. Quagliarella (CIRA) and M. Vasile (University of Strathclyde).
 - *‘Applications of optimization in engineering design automation’*. Organized by D. Ertner and T. Prante (V-Research), M. Affenzeller (University of Applied Science Upper Austria), J. Johansson (Jönköping University) and W. J. C. Verhagen (Delft University of Technology).
 - *‘Strategic interaction: theoretical and computational questions of Optimization and Game Theory’*. Organized by C. de Nicola and L. Mallozi (University of Naples Federico II).

- *‘Miscellaneous of applications of Evolutionary Algorithms in Energy and Fall prediction’*. Organized by A. Brunete, M. Hernando and E. Gambao (UPM) and Diego Oliva (Tecnológico de Monterrey)
- One (1) Special Technological Session (STS) on *‘Advanced Design Optimization and Control Challenges of new aircraft/engines configurations for future subsonic, transonic and supersonic transportation’*. Organized by J. Periaux (CIMNE) and D. Redondo (Airbus).
- Registered attendants by nationalities (not including plenary speakers and local organizing committee members): 13 Spain, 17 Germany, 10 Italy, 18 United Kingdom, 8 France, 4 Japan, 8 Belgium, 7 Austria, 3 Greece, 2 Portugal, 1 Iran, 1 Mexico, 1 Brazil, 2 Sweden, 1 Czech Republic and 2 USA.

Among the 92 presentations of the EUROGEN 2017 conference, 35 extended papers were selected for publication in this volume after peer-review by members of the European Scientific Programme Committee.

The Scientific Organizing Committee and the Local Organizing Committee acknowledge the sponsorship of the following organizations through financial support or/and assistance during the development of the event: Research projects E-CAERO and SSEID, Center for Computational Simulation, AIRBUS, ANSYS, NUMECA, ESTECO and Fundación Marqués de Suances.

Finally, the two Committees above are grateful to all the members of the European Scientific Committee, the European Technical Committee and the International Corresponding members.

Madrid, Spain
 Madrid, Spain
 Barcelona, Spain
 Kaiserslautern, Germany
 Capua, Italy
 Athens, Greece

Esther Andrés-Pérez
 Leo M. González
 Jacques Periaux
 Nicolas Gauger
 Domenico Quagliarella
 Kyriakos Giannakoglou

Contents

Part I Adjoint Methods for Optimisation, Mesh Adaptation and Uncertainty Quantification	
Gradient Projection, Constraints and Surface Regularization Methods in Adjoint Shape Optimization	3
Pavlos P. Alexias and Eugene de Villiers	
Adjoint Shape Optimisation Using Model Boundary Representation	19
Marios Damigos and Eugene de Villiers	
CAD and Adjoint Based Multipoint Optimization of an Axial Turbine Profile	35
Ismael Sanchez Torreguitart, Tom Verstraete and Lasse Mueller	
A Comparative Study of Two Different CAD-Based Mesh Deformation Methods for Structural Shape Optimization	47
Marc Schwalbach, Tom Verstraete, Jens-Dominik Müller and Nicolas Gauger	
Node-Based Adjoint Surface Optimization of U-Bend Duct for Pressure Loss Reduction	61
G. Alessi, L. Koloszar, Tom Verstraete and J. P. A. J. van Beeck	
On the Properties of Solutions of the 2D Adjoint Euler Equations	77
Carlos Lozano	
Finite Transformation Rigid Motion Mesh Morpher	93
Athanasios G. Liatsikouras, Guillaume Pierrot, Gabriel Fougeron and George S. Eleftheriou	
The Unsteady Continuous Adjoint Method Assisted by the Proper Generalized Decomposition Method	109
V. S. Papageorgiou, K. D. Samouchos and Kyriakos Giannakoglou	

A Two-Step Mesh Adaptation Tool Based on RBF with Application to Turbomachinery Optimization Loops	127
Flavio Gagliardi, Konstantinos T. Tsiakas and Kyriakos Giannakoglou	
Adjoint-Based Aerodynamic Optimisation of Wing Shape Using Non-uniform Rational B-Splines	143
Xingchen Zhang, Rejish Jesudasan and Jens-Dominik Müller	
Part II Surrogate-Assisted Optimization of Real World Problems	
A Comparative Evaluation of Surrogate Models for Transonic Wing Shape Optimization	161
Emiliano Iuliano	
Study of the Influence of the Initial a Priori Training Dataset Size in the Efficiency and Convergence of Surrogate-Based Evolutionary Optimization	181
Daniel González-Juarez and Esther Andrés-Pérez	
Garteur AD/AG-52: Surrogate-Based Global Optimization Methods in Preliminary Aerodynamic Design	195
Esther Andrés-Pérez, Daniel González-Juarez, Mario Martin, Emiliano Iuliano, Davide Cinquegrana, Gerald Carrier, Jacques Peter, Didier Bailly, Olivier Amoignon, Petr Dvorak, David Funes, Per Weinerfelt, Leopoldo Carro, Sancho Salcedo, Yaochu Jin, John Doherty and Handing Wang	
A Response Surface Based Strategy for Accelerated Compressor Map Computation	211
Dmitrij Ivanov, Dieter Bestle and Christian Janke	
Surrogate-Based Shape Optimization of the ERCOFTAC Centrifugal Pump Impeller	227
Remo De Donno, Stefano Rebay and Antonio Ghidoni	
CFD Based Design Optimization of a Cabinet Nitrogen Generator	247
Bárbara Arizmendi Gutiérrez, Edmondo Minisci and Greig Chisholm	
Delaunay-Based Global Optimization in Nonconvex Domains Defined by Hidden Constraints	261
Shahrouz Ryan Alimo, Pooriya Beyhaghi and Thomas R. Bewley	

Part III Applications of Optimization in Engineering Design Automation

Optimized Vehicle Dynamics Virtual Sensing Using Metaheuristic Optimization and Unscented Kalman Filter 275
 Manuel Acosta and Stratis Kanarachos

Optimization of Ascent Assembly Design Based on a Combinatorial Problem Representation 291
 Michael Hellwig, Doris Entner, Thorsten Prante, Alexandru-Ciprian Zăvoianu, Martin Schwarz and Klara Fink

On the Optimization of 2D Path Network Layouts in Engineering Designs via Evolutionary Computation Techniques 307
 Alexandru-Ciprian Zăvoianu, Susanne Saminger-Platz, Doris Entner, Thorsten Prante, Michael Hellwig, Martin Schwarz and Klara Fink

Taking Advantage of 3D Printing So as to Simultaneously Reduce Weight and Mechanical Bonding Stress 323
 Markus Schatz, Robert Schweikle, Christian Lausch, Michael Jentsch and Werner Konrad

Interactive Optimization of Path Planning for a Robot Enabled by Virtual Commissioning 339
 Ruth Fleisch, Doris Entner, Thorsten Prante and Reinhard Pfefferkorn

Box-Type Boom Design Using Surrogate Modeling: Introducing an Industrial Optimization Benchmark 355
 Philipp Fleck, Doris Entner, Clemens Münzer, Michael Kommenda, Thorsten Prante, Martin Schwarz, Martin Hächl and Michael Affenzeller

Knowledge Objects Enable Mass-Individualization 371
 Joel Johansson and Fredrik Elgh

Free-Form Optimization of A Shell Structure with Curvature Constraint 387
 Masatoshi Shimoda and Kenichi Ikeya

Application of Game Theory and Evolutionary Algorithm to the Regional Turboprop Aircraft Wing Optimization 403
 Pierluigi Della Vecchia, Luca Stingo, Fabrizio Nicolosi, Agostino De Marco, Elia Daniele and Egidio D’Amato

Industrial Application of Genetic Algorithms to Cost Reduction of a Wind Turbine Equipped with a Tuned Mass Damper 419
 Jordi Pons-Prats, Marti Coma, Jaume Betran, Xavier Roca and Gabriel Bugada

Part IV Optimization Under Uncertainty

Aerodynamic Shape Optimization by Considering Geometrical Imperfections Using Polynomial Chaos Expansion and Evolutionary Algorithms	439
---	------------

Athanasios G. Liatsikouras, Varvara G. Asouti,
Kyriakos Giannakoglou and Guillaume Pierrot

Multiobjective Optimisation of Aircraft Trajectories Under Wind Uncertainty Using GPU Parallelism and Genetic Algorithms	453
---	------------

Daniel González-Arribas, Manuel Sanjurjo-Rivo and Manuel Soler

Multi-objective Optimization of A-Class Catamaran Foils Adopting a Geometric Parameterization Based on RBF Mesh Morphing	467
---	------------

Marco Evangelos Biancolini, Ubaldo Cella, Alberto Clarich
and Francesco Franchini

Development of an Efficient Multifidelity Non-intrusive Uncertainty Quantification Method	483
--	------------

Saeed Salehi, Mehrdad Raisee, Michel J. Cervantes
and Ahmad Nourbakhsh

Part V Multi-disciplinary Design Optimization

Evolving Neural Networks to Optimize Material Usage in Blow Molded Containers	501
--	------------

Roman Denysiuk, Fernando M. Duarte, João P. Nunes
and António Gaspar-Cunha

Coupled Subsystem Optimization for Preliminary Core Engine Design	513
--	------------

Simon Extra, Michael Lockan, Dieter Bestle and Peter Flassig

Progresses in Fluid-Structure Interaction and Structural Optimization Numerical Tools Within the EU CS RIBES Project	529
---	------------

Marco Evangelos Biancolini, Ubaldo Cella, Corrado Groth,
Andrea Chiappa, Francesco Giorgetti and Fabrizio Nicolosi

Part I
**Adjoint Methods for Optimisation, Mesh
Adaptation and Uncertainty Quantification**

Gradient Projection, Constraints and Surface Regularization Methods in Adjoint Shape Optimization



Pavlos P. Alexias and Eugene de Villiers

Abstract This paper deals with the treatment of various problems that are present in adjoint-based shape optimization applications, in which a parameterization of the surface is absent. A general implicit smoothing algorithm is used to reduce high-frequency noise which might be present in the gradients that are calculated using a continuous adjoint solver. The implicit smoother allows the definition of patches on the shape that need to remain fixed during shape optimization and automatically secures surface gradient continuity between constrained and deformable patches. Along with the gradient smoothing, a surface mesh regularization algorithm is presented and used to support high-quality elements and mesh uniformity during each optimization step. In the end, the capability and the effectiveness of the method are demonstrated in various industrial test cases.

Introduction

In the context of gradient-based numerical optimization, the adjoint method is the most cost-effective way to calculate the gradients of an objective function with respect to the design variables. This is due to the fact that the cost of the adjoint gradient calculation is practically independent of the number of the design variables of the optimization problem (Jameson 1988, 1995; Pironneau 1974). Design variable independence allows the exploration of richer design spaces and consequently convergence to better optimum solutions without increasing the computational cost. In gradient-free methods, like evolutionary algorithms (EA), the curse of dimensionality limits optimization w.r.t. many design variables due to significant cost increases.

P. P. Alexias (✉)
Engys S.R.L, Via del Follatoio 12, 34148 Trieste, Italy
e-mail: p.alexias@engys.com

E. de Villiers
Engys Ltd, Studio 20, RVPB, John Archer Way, London SW 18 3SX, UK
e-mail: e.devilliers@engys.com

An important aspect of shape optimization applications is the selection of the design space and, accordingly, the selection of the design variables. A rational choice would be to keep the shape's design parameters as design variables, but in CAD-free applications the existence of an analytical parametrization of the surface is not available for most applications. To deal with this problem, there are two main approaches to follow: (1) free-form deformation (FFD); in which the shape is enclosed by a hull object (e.g. a lattice) and (2) the node-based approach; in which each surface node of the computational mesh is considered as a design variable. In free-form deformation the shape's deformation is translated into the deformation of the hull based on interpolation functions. There are various free form deformation techniques in the literature which are utilized in shape optimization problems. They use several types of parametrization basis functions to describe the hull object such as B-Splines, NURBS (Martin et al. 2014), Radial Basis Functions (RBFs) (Kiachagias et al. 2015) or Harmonic Coordinates (Joshi et al. 2007). A notable drawback of the FFD methods is that the optimum solution is affected strongly by the user input. A different choice of geometry for the hull object or a different number of design parameters may result in different solutions, which can negatively impact the effectiveness of the method and reduce its general utility.

In the absence of parametrisation, the surface nodes of the CFD mesh can be used as design variables. The latter approach offers the richest possible design space (for the given spatial discretization) allowing the generation of better optimum solutions and the implementation of high-complexity constraints. However, any numerical noise in the adjoint derivatives, combined with the fact that each surface node is being perturbed independently from its neighbours, can create oscillations and irregularities. To avoid reducing the smoothness of the surface and negatively impacting the convergence of the optimization problem it is necessary to generate a smooth representation of the gradient. In FFD methods this smoothing takes place naturally through the given parametrization projecting the gradients from the mesh surface onto the parametric subspace. In node-based approaches the most established methods are the implicit smoothing technique (Jameson and Vassberg 2000), also known as Sobolev gradient smoothing, and an explicit technique which uses Gaussian filter kernels (Stück and Rung 2011).

In the present paper, the implicit technique will be used, in a modified form, that provides the advantages of Stück's and Rung's explicit technique in terms of translating the smoothing parameters into meaningful design variables. In addition, a mesh surface regularization algorithm will be presented and used effectively to prevent the degeneration and the appearance of low quality elements caused by big surface deformations. The gradients computed by the adjoint method are w.r.t. the point normal directions of the surface, making it necessary to re-order the points in the tangent direction to keep the mesh's surface elements uniformity and quality. The algorithm is based on the maximization of a mesh quality metric using analytical derivative expressions of geometric mesh quantities.

The Continuous Adjoint Method

A brief overview of the continuous adjoint method (Karpouzas et al. 2016; Papoutsis-Kiachagias and Giannakoglou 2014) will be presented in this section. Without loss of generality, assuming a laminar flow of an incompressible fluid, the Navier-Stokes equations can be written as,

$$R^p = -\frac{\partial v_j}{\partial x_j} = 0 \tag{1}$$

$$R_i^v = v_j \frac{\partial v_i}{\partial x_j} - \frac{\partial}{\partial x_j} \left[\nu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] + \frac{\partial p}{\partial x_i} = 0 \quad i = 1, 2, 3 \tag{2}$$

where v_i is the velocity in direction of the Cartesian coordinates, ν is the kinematic viscosity and p is the static pressure field divided by the fluid density ρ .¹ Defining a general objective function that contains both surface and volume integrals as

$$F = \int_S F_s dS + \int_\Omega F_\Omega d\Omega \tag{3}$$

then F is augmented by the state equations leading to

$$F_{aug} = F + \int_\Omega u_i R_i^v d\Omega + \int_\Omega q R^p d\Omega \tag{4}$$

Here, u_i and q are the adjoint variables of the flow velocity and the static pressure respectively. Note that since the Navier-Stokes equations are satisfied, it holds that $F_{aug} = F$. Next comes the differentiation of the augmented objective function using the Green-Gauss theorem to pass from volume integrals to surface integrals. Zeroing the partial derivatives of flow field values results in the adjoint equations and adjoint boundary conditions:

$$R^q = \frac{\partial u_i}{\partial x_i} - \frac{\partial F_\Omega}{\partial p} = 0 \tag{5}$$

$$R_i^u = -v_j \frac{\partial u_i}{\partial x_j} + u_j \frac{\partial v_j}{\partial x_i} - \nu \frac{\partial}{\partial x_j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \frac{\partial q}{\partial x_i} + \frac{\partial F_\Omega}{\partial v_i} = 0 \tag{6}$$

After solving the adjoint equations, in similar way with the primal Navier-Stokes equations, the final sensitivities values w.r.t. the design variables b_n are,

$$G = \frac{\delta F_{aug}}{\delta b_n} = - \int_{S_w} \left[\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - q n_i \right] \frac{\partial v_j}{\partial x_k} \frac{\partial x_k}{\partial b_n} dS \tag{7}$$

¹On the aforementioned equations and on what follows, the Einsteins summation convention applies for the lower-case indices, unless declared differently.

Implicit Smoothing

The absence of a parametric description for the geometry or the existence of a low-quality computational mesh or partially converged primal solution (Eq. 1), may result in the intrusion of noise into the adjoint sensitivities. In the node-based framework, the surface sensitivities are translated directly to node displacements and this will transfer the noise from the sensitivities on the surface. This noise intrusion can lead to impractical, non-manufacturable shapes and make our optimization problem difficult to converge (or even diverge). It is thus necessary to create a smooth gradient representation to cut-off any undesired oscillation.

Implicit Smoothing Equation

Based on the Sobolev gradient projection introduced by Jameson (1988), the smoothed gradient field \bar{G} is calculated from the initial gradient G through the diffusion-like equation

$$\bar{G} - \varepsilon \nabla^2 \bar{G} = G \quad (8)$$

where ε is the smoothing coefficient which defines the smoothness of the final gradient representation. This generates a free variable for the optimization problem and thus the choice of the smoothing coefficient is case-specific. There are various suggestions in the literature for an optimum choice of ε (Schmidt et al. 2008; Gherman and Schulz 2005). In the present paper, based on the work of Stück and Rung (2011), in which they use an equivalent explicit technique, the coefficient ε will be translated into a maximum allowed oscillation radius on the shape's surface. This can be done by taking the fundamental solution of the diffusion equation of a point source (for 2 variables)

$$\Phi(x, \varepsilon) = \frac{1}{4\pi\varepsilon} e^{(-x^2/4\varepsilon)} \quad (9)$$

where it can be noticed that the above equation is identical to the Gauss function with a standard deviation of $\sigma = \sqrt{2\varepsilon}$. This, allows the definition of a smoothing radius equivalent to three times the standard deviation of the Gauss bell in which every oscillation with a radius smaller than the defined will be severed.

Solving an elliptic equation implicitly will result in the spreading of sensitivity information inside the whole computational domain. However, only 0.3% of the information from a point source will pass outside of the 3σ smoothing radius. For all intents and purposes the smoothing will thus be acting locally inside the predefined radius.

Constraints and Continuity

Smoothing the sensitivities through the solution of a PDE allows a straight-forward definition of constrained areas on the optimized shape. Zeroing the sensitivities on the non-moving patches and defining a fixed value boundary condition on the edges between deformable and constrained patches will result in having non-zero sensitivities only on the deformable patches. Even though applying a fixed value boundary condition on the common edges will prevent the smoother from generating sensitivities on constrained patches, there is no control on the way the shape transits from a constrained to a deformed patch. In cases where there are big sensitivity magnitudes in the vicinity of common edges, strong discontinuities will arise. Since it is important to maintain a smooth transition between the constrained and the deformable patches, the sensitivities with a geodesic distance from the fixed boundaries smaller than half of the smoothing radius are set to zero. In this way, the points close to the transition zone will be exactly on the Gauss bell curve maintaining C_2 continuity. The geodesic distance calculation is performed by solving the eikonal equation

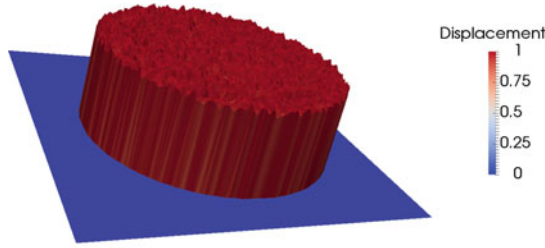
$$|\nabla\phi(x)| = \frac{1}{f(x)}, \quad x \in \Omega \quad (10)$$

subject to boundary condition $\phi|_{\partial\Omega}$. Solving for $\phi(x)$ will result in the shortest time needed to travel from the boundary $\partial\Omega$ to any point x inside Ω with a velocity $f(x)$. In the special case where $f(x) = 1$ the solution $\phi(x)$ is the shortest distance of x from $\partial\Omega$.

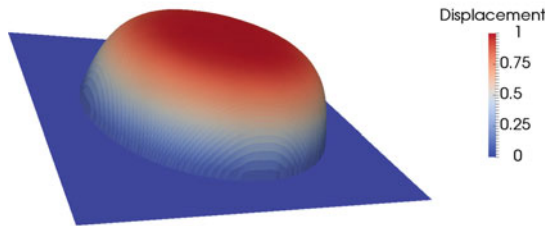
The effect and the importance of securing a smooth transition is demonstrated in Fig. 1. Smoothing is performed on a uniform displacement with superimposed noise (Fig. 1a), while respecting a circular constraint. Solving the Eq. 8 without transition treatment, will result on the shape depicted in Fig. 1b which does not have a continuous surface gradient at the common edge. Figure 1c demonstrates how a smooth transition can be achieved at the constrained boundary by zeroing the displacement values in elements with a distance from the constraint patch smaller than half of the smoothing radius.

Numerical Implementation

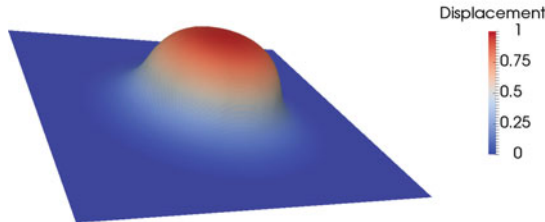
In the present paper, the code, the solvers and the applications have been developed under the open source framework of OPENFOAM. The sensitivity smoothing pertains to the solution of a two-dimensional partial differential equation (PDE) in a three-dimensional surface. In the case of a homogenous surface mesh with known topology, the equation can be solved moving between the Euclidian and the curvilinear space through covariant and contravariant functions. In the general case of unstructured grids this method will not work without obtaining an implicit surface representation (Bertalmio et al. 2001). In the present paper, a different approach will be followed



(a) Uniform distribution with additive noise.



(b) Implicit smoothing equation solution applying zero fixed value boundary conditions on the common edges of deformable and constrained patch.



(c) Implicit smoothing equation solution with a smooth transition.

Fig. 1 Differences in the shapes when solving the implicit smoothing equation (Eq. 8) with (c) and without (b) ensuring a smooth transition from the constrained to the deformable patch

that allows the solution of any PDE via the so called finite area approach. The method is implemented in the same way as the finite volume method in OPENFOAM, with the difference that the discretized elements are polygonal faces with individual local coordinate systems. The surfaces curvature enters the calculations through tensor transformations of every face centroid into a global coordinate system (Tukovic and Jasak 2009).

Surface Mesh Regularization

In the node-based method, the gradients of the objective function are computed w.r.t. the normal directions of the nodes on the surface. Moving continuously towards the normal direction, even with a smooth gradient representation, will eventually lead to the generation of low-quality surface mesh elements, tangled faces or high non-uniformity. To avoid this, a separate movement in the tangent direction is necessary for the restoration of the mesh quality.

In this section a mesh optimization algorithm aimed at the maximization of an element-wise quality metric will be presented. The goal is to calculate the new set of surface node positions that maximize this metric, assuming that the movement of the surface nodes in the tangent direction will not have a significant impact. The aforementioned assumption can be rationalised in the context of iterative surface motion, where the point locations are updated repeatedly.

The quality metric is defined for every surface element as

$$\mu = \alpha \frac{S_e}{P_e^2} \quad (11)$$

where S_e is the surface of the element, P_e is the perimeter and α is a normalization factor. The goal is to reposition the vertices of the surface mesh such that the faces will have the maximum metric value. In order to do so, it is necessary to calculate the derivative of the metric w.r.t. the position of the points that constitute a face. An extensive analysis of the differentiation of such kind of quality metrics goes beyond the scope of this paper. For a detailed description on the differentiation of geometric quantities in discrete geometry refer to Alexias and De Villiers (2016).

If a point \mathbf{P} on the surface, is surrounded by N faces, the total derivative of that point is the arithmetic mean of the sensitivities from all contributing faces.

$$\delta = \frac{1}{N} \sum_{n=1}^N \frac{d\mu}{d\mathbf{P}} \quad (12)$$

Having obtained the sensitivity derivatives of the objective function, an optimization step is performed through a quasi-Newton method as

$$\mathbf{P}^{new} = \mathbf{P}^{old} + \lambda H^{-1} \cdot \delta \quad (13)$$

where H is the Hessian matrix approximation using a limited memory BFGS method Nocedal (1980).

An example of the surface mesh regularization algorithm is demonstrated in Fig. 2 on a simple surface deformation case. Applying a displacement field on a flat surface may result in the creation of a surface mesh with distorted and anisotropic elements. Using mesh regularisation during the deformation results in a higher quality mesh with greater uniformity. The impact of the optimization algorithm on the mesh quality

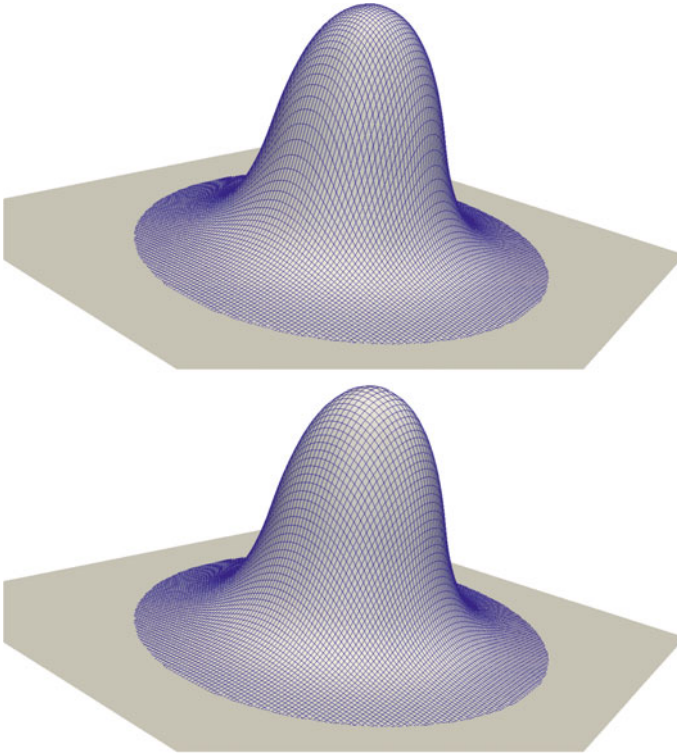


Fig. 2 Surface mesh without (top) and with (bottom) mesh regularization after the surface displacement

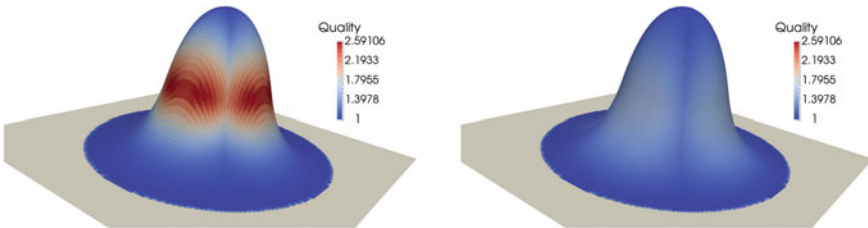


Fig. 3 Aspect ratio of surface mesh elements before (left) and after (right) mesh regularization

can be independently quantified by examining an alternative mesh quality metric, like aspect ratio. Figure 3 depicts the aspect ratio before and after the regularization. Taking into account that the optimum aspect ratio value is one, it is clear that the optimisation procedure significantly improves this metric, leading to a higher quality surface mesh.

Applications

In this section, the algorithms and the methods, presented in this paper, are combined to achieve a fully automated optimization process with minimal user intervention. The efficacy of the method will be demonstrated in two large-scale industrial test cases consisting of internal and external flow respectively.

Power Losses Minimization

The first application aims at the minimization of the power losses of an air S-bend duct. The flow is laminar with a Reynolds number of 350, having a structure computational mesh comprising 700 thousand cells.

Figure 4 illustrates the duct geometry and the parts of the duct that are allowed to move during the shape optimization. The objective function subject to minimization is

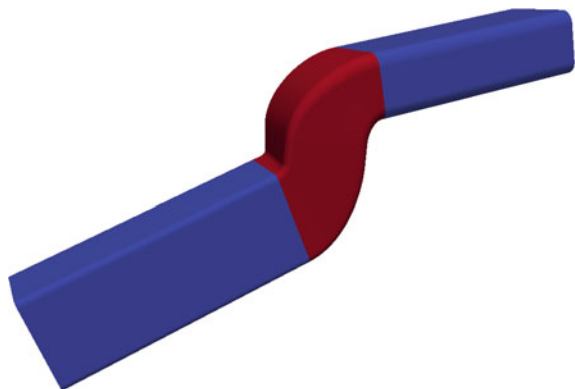
$$F = \int_S (p + \frac{1}{2}v_i^2)v_i n_i dS \tag{14}$$

Following a steepest decent optimization method every new position of the surface nodes is given by

$$\mathbf{x}^i = \mathbf{x}^{i-1} - a\overline{\mathbf{G}} \cdot \mathbf{n} \tag{15}$$

where $\overline{\mathbf{G}}$ is the smoothed gradient, a is the step of the steepest decent and \mathbf{n} is the normal direction of each point. Figure 5 illustrates the differences between smoothed and raw gradients when applying a smoothing radius of 1 cm. The smoothed gradients satisfy the constraints and simultaneously allow smooth transition between the constrained and unconstrained patches.

Fig. 4 Geometry of the S-bend duct. With red color is the deformable surface while the rest remains fixed



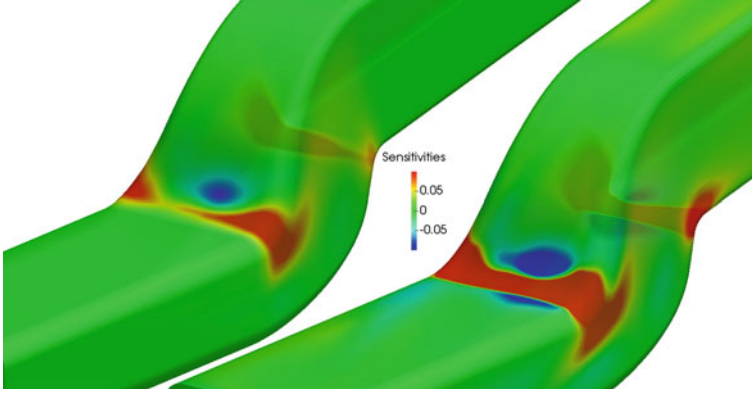


Fig. 5 Comparison between the smoothed (left) and the raw (right) gradients applying a smoothing radius of 1cm

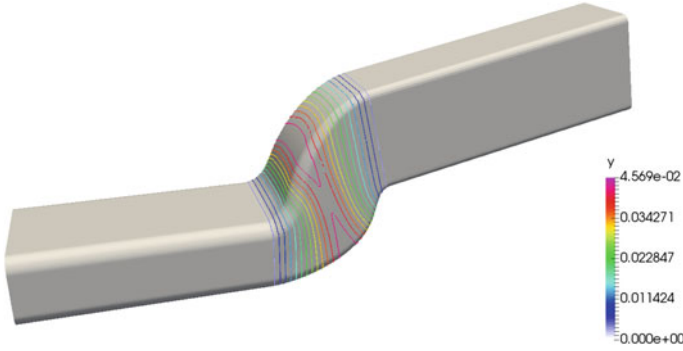


Fig. 6 Geodesic curves on the 3D surface defining the shortest curvilinear distance of any point from the constrained boundaries

Solving the eikonal equation will result in the calculation of the geodesic distance curves on the 3D deformable surface. To ensure a smooth transition a zero-band zone needs to be created and thus the points belonging to a distance smaller than half the smoothing radius (0.5 cm in the present case) will be assigned a zero gradient value. Figure 6 illustrates the geodesic distances on the deformable surface of the S-bend duct.

After performing the steepest decent optimization step, the surface optimization takes place: improving the surface mesh quality and maintaining the mesh uniformity. Thus, the final node locations are calculated through

$$\mathbf{x}_{final}^i = \mathbf{x}_i + \delta_{total} \quad (16)$$

where δ_{total} is the total displacement resulting from the mesh optimization algorithm. Figure 7 illustrates the difference between having a smooth transition with a surface

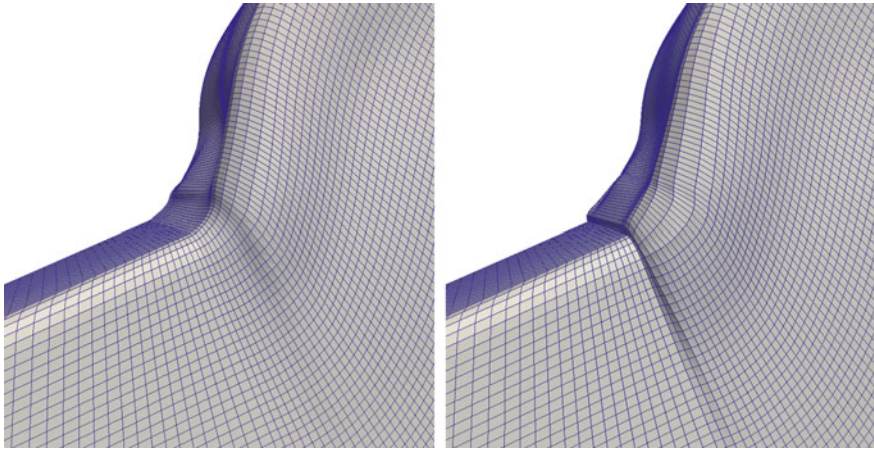


Fig. 7 Differences between deformed shape using smooth transition and mesh optimization (left) and using only gradient smoothing (right)

mesh optimization and simplistic gradient smoothing. In the first case, the mesh has improved uniformity and higher quality faces. Most importantly, there is no “step” in the surface at the interface. This “step” typically has a negative impact on the optimization problem as it creates inferior quality elements that hamper the convergence of the CFD solution (Fig. 8).

Beyond the surface displacement algorithm, it is necessary to adapt the internal mesh points for re-solving the CFD equations. For this purpose, a Laplacian equation with an inverse distance diffusion coefficient is solved combined with a mesh optimization approach that guarantees a high-quality mesh, even during extreme deformations.

Figure 9 displays the pressure losses reduction history as a function of the optimization cycles. It can be seen that after 35 optimization steps the pressure losses were reduced by 17.1%. It is noticeable that on every optimization cycle the mesh quality remained exceptionally high allowing us to proceed through a series of complex shapes and deformations without the need of remeshing. This contributes a lot to the automation of the optimization procedure and the reduction of the computational time.

Drag Force Minimization

The second application aims at the minimization of the drag force of the DrivAer car model (Heft et al. 2012). In detail, the fast-back configuration with a smooth underbody, with mirrors and wheels is used in this application. Only half of the car geometry is meshed and used for the simulation, with a computational grid

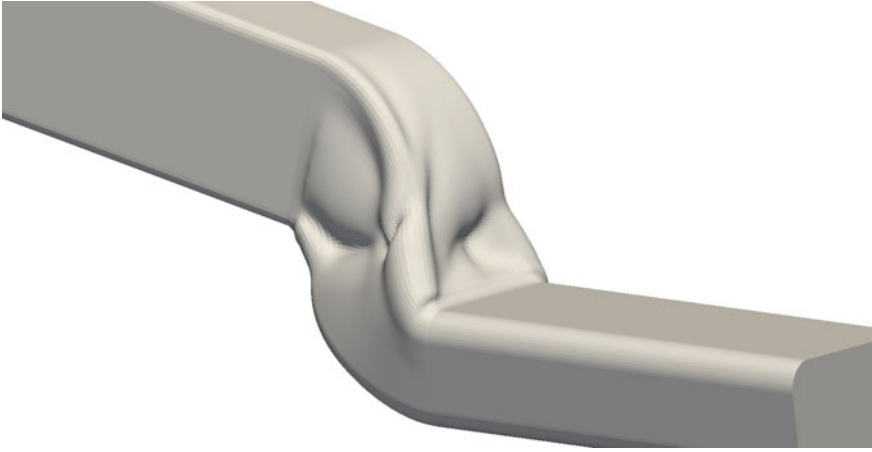


Fig. 8 Optimized sBend duct shape, achieving a 17.1% power losses reduction

Fig. 9 Pressure losses reduction for every optimization step



comprising around 6 million cells. The flow is turbulent and modelled with the Spalart-Allmaras turbulence model with wall functions. Even though the flow around a car does not reach a steady state solution, a time invariant CFD model is used for the simulation. This simplifies the optimization procedure, avoiding barriers and difficulties which arise when we are dealing with unsteady adjoint equations.

As it can be seen from Fig. 11, illustrated with green color, only a portion of the rear part of the car is allowed to move during the optimization. After 8 optimization cycles using a steepest decent method the algorithm converged to a shape with a reduction in drag of more than 2% (Fig. 10). Figure 12 demonstrates the displacement field (after applying the smoothing algorithm) for the first optimization cycle. It can be seen that the highest deformation lies on a thin strip on the rear part of the trunk of the car. Observing the Fig. 13, which compares the initial and the optimum shape, the generation of a spoiler, by lowering the trunk, leads to an increased pressure on the

Fig. 10 Convergence history of the drag force for every optimization cycle. A reduction of more than 2% can be observed

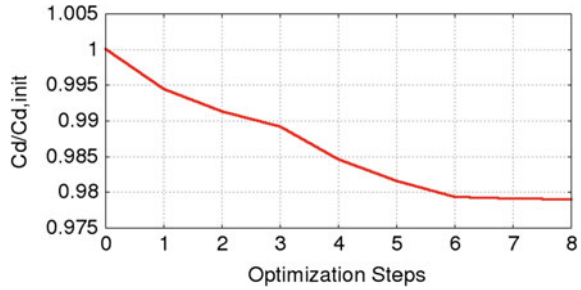
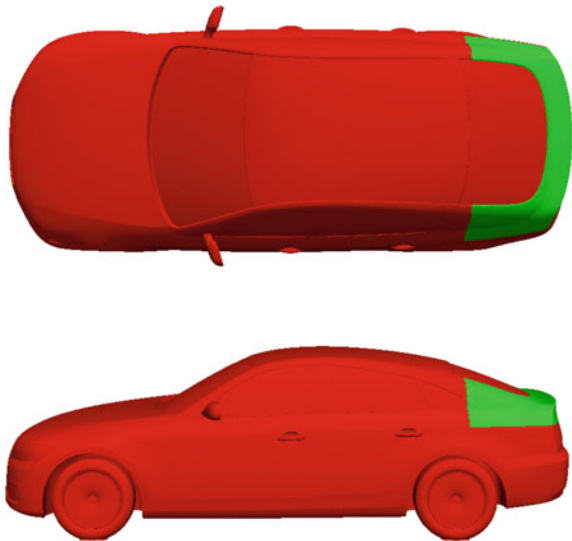


Fig. 11 DrivAer car geometry. With green color is the deformable part while the rest has to remain fixed during the optimization



rear part of the car contributing to the reduction of the drag force. This percentage of drag reduction (2.1%) may seem small but is significant considering that only a small portion of the rear car was subject to shape deformation during the optimization.

Conclusions

In this paper, a complete and fully automated framework has been developed to deal with gradient and surface treatment problems in the context of adjoint-based optimization. The gradients of an objective function w.r.t. the mesh surface nodes are computed using the continuous adjoint method. Those gradients are smoothed via an implicit solver that removes any unnecessary oscillation and noise. Using the same smoothing framework in conjunction with geodesic distances resulted in the proper application of constraints while keeping a desirable level of continuity. As a

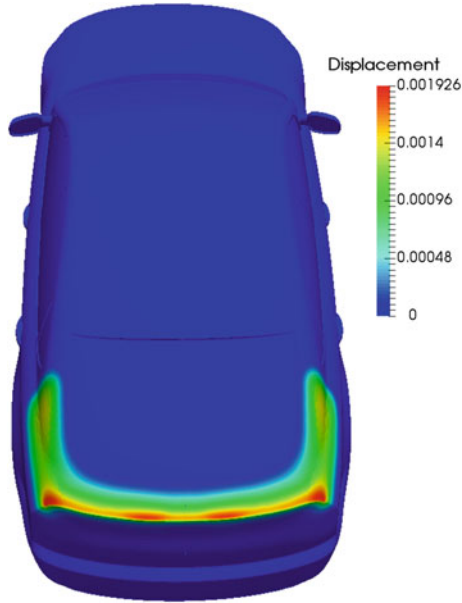


Fig. 12 Smooth displacement field during the first optimization step

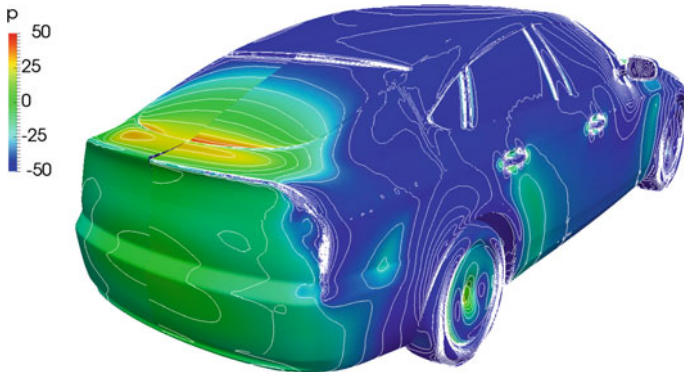


Fig. 13 Initial (left) and final (right) shape after 8 optimization cycles together with the pressure distribution. Lowering the trunk leads to an increased pressure on the rear part of the car which contributes to the reduction of the drag force

final step, a quality based optimisation was performed on the surface. The methods efficacy was demonstrated by achieving significant performance improvements for both internal and external flow test cases.

Acknowledgements This work has been conducted within the IODA project (<http://ioda.sems.qmul.ac.uk>), funded by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

References

- Alexias P, De Villiers E (2016) Sphericity: mesh optimization for arbitrary element topology. In: VII ECCOMAS Congress, 5–10 June 2016, Crete Island, Greece
- Bertalmio M, Cheng LT, Osher S, Sapiro G (2001) Variational problems and partial differential equations. *J Comput Phys* 174(2):759–780, TBW
- Gherman I, Schulz V (2005) Preconditioning of one-shot pseudo-timestepping methods for shape optimization. *PAMM Proc Appl Math Mech* 5(1):741–742
- Heft A, Indinger T, Adams N (2012) Experimental and numerical investigation of the DrivAer model. ASME (2012) symposium on issues and perspectives in automotive flows. Puerto Rico, USA, pp 41–51
- Jameson A (1995) Optimum aerodynamic design using CFD and control theory. AIAA-1995-1729-CP
- Jameson A (1988) Aerodynamic design via control theory. *J Sci Comput* 3(3):233–260
- Jameson A, Vassberg JC (2000) Studies of alternative numerical optimization methods applied to the brachistochrone problem. *Comput Fluid Dyn* 9:281–296
- Joshi P, Meyer M, Derose T, Green B, Sanocji T (2007) Harmonic coordinates for character articulation. *ACM Trans Graph* 26(3):71
- Karpouzias Georgios K, Papoutsis-Kiachagias EM, Thomas S, de Villiers E, Giannakoglou KC, Othmer C (2016) Adjoint optimization for vehicle external aerodynamics. *Int J Automot Eng* 7(1):1–7
- Martin MJ, Andres E, Lozano C, Valero E (2014) Volumetric B-splines shape parametrization for aerodynamic shape design. *Aerosp Sci Technol* 37:2636
- Nocedal J (1980) Updating quasi-Newton matrices with limited storage. *Math Comput* 35:773–782
- Papoutsis-Kiachagias E-M, Porziani S, Groth C, Biancolini ME, Costa E, Gianakoglou KC (2015) Aerodynamic optimization of car shapes using the continuous adjoint method and an RBF morpher. In: EUROGEN 2015, Glasgow, UK, September 14–16
- Papoutsis-Kiachagias E, Giannakoglou K (2014) Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. *Arch Comput Methods Eng*. <https://doi.org/10.1007/s11831-014-9141-9>
- Pironneau O (1974) On optimum design in fluid mechanics. *J Fluid Mech* 64(1):97–110
- Schmidt S, Ilic C, Gauger N, Schulz V (2008) Shape gradients and their smoothness for practical aerodynamic design optimization. Preprint-Number SPP1253-10-03
- Stück A, Rung T (2011) Filtered Gradients for adjoint-based shape optimisation. In: 20th AIAA computational fluid dynamics conference 27–30 June 2011, Honolulu, Hawaii
- Tukovic Z, Jasak H (2009) Simulation of thin liquid film flow using OpenFOAM finite area method. In: 4th OpenFOAM workshop, Montreal, Canada

Adjoint Shape Optimisation Using Model Boundary Representation



Marios Damigos and Eugene de Villiers

Abstract Manipulating CAD geometry using primitive components rather than the originating software is typically a challenging prospect. The parameterization used to define the geometry of a model is often integral to the efficiency of the design. However, it is not always possible to access these parameters due to the closed-source, non-standardized nature of most CAD software. A sensible choice, is to use standard CAD files which have an open format, in order to read a model. Importing such a file gives access to the Boundary Representation (BRep) of the model and consequently its boundary surfaces which are usually trimmed patches. Therefore, in order to connect Adjoint optimization to the industrial design framework (CAD) in a generic manner, the BRep must be used as a means of changing a model's shape. In this study, Geometry Morphing, a method of imposing up to C1 continuity between moving BRep patches is demonstrated and then applied to various optimization cases.

Introduction

One of the biggest challenges in modern day CFD and optimization is to establish the missing connection with industrial design. Each CAD software has its own proprietary format and parameterization, which is typically not disclosed by the vendor. Due to the closed source nature of the most popular CAD packages, the above mentioned formats cannot be accessed. Consequently, one must use an alternative way to access a model's information. A common choice is to use standard CAD formats such as STEP or IGES (Nowacki and Dannenberg 1986) which contain the Boundary Representation of a geometric model.

M. Damigos (✉)
ENGYS Srl., Via del Follatoio, 12 34148 Trieste, Italy
e-mail: m.damigos@engys.com

E. de Villiers
ENGYS Ltd, Studio 20, Royal Victoria Patriotic Building,
John Archer Way, SW18 3SX, London, UK
e-mail: e.devilliers@engys.com

Boundary Representation (BRep)

Boundary Representation (Stroud 2006; Mantyla 1988) is a method for representing shapes in solid modeling. A solid is represented by the BRep format using surface elements defining the interface between solid and non-solid volumes. The BRep format is composed of two parts: topological data and geometry. The topology of a BRep is created using vertices, edges, faces, shells and ultimately solids. Regarding their underlying geometry:

1. Vertices: The underlying geometry of a vertex is simply a 3D point.
2. Edges: Edges are curves bounded by the points describing their boundary vertices. In the general case, the geometry of an edge is only a segment of its underlying curve, since the topological bounds of an edge and the geometrical bounds of a curve are not strictly identical.
3. Faces: Similarly to edges, faces are described by surfaces bounded by a closed loop of edges. Moreover, the geometry of a face is, in general, a part of its underlying surface, since its boundary loop of edges does not coincide with the natural bounds of the surface.
4. Shells: A shell is composed of multiple faces connected to each other and has no particular underlying geometry.
5. Solids: A solid similarly to a shell, does not have an underlying geometry and is practically the volume bounded by a collection of shells.

The mathematical description of curve and surface elements of a BRep model could vary. Elementary curves or surfaces such as circular arcs, planes or cylinders for example, could be stored explicitly. However, more complex elements are stored in parametric form, most commonly NURBS (Piegl and Tiller 1995). The conversion from elementary curves or surfaces to NURBS is trivial, thus the BRep geometry will be handled as NURBS geometry for uniformity.

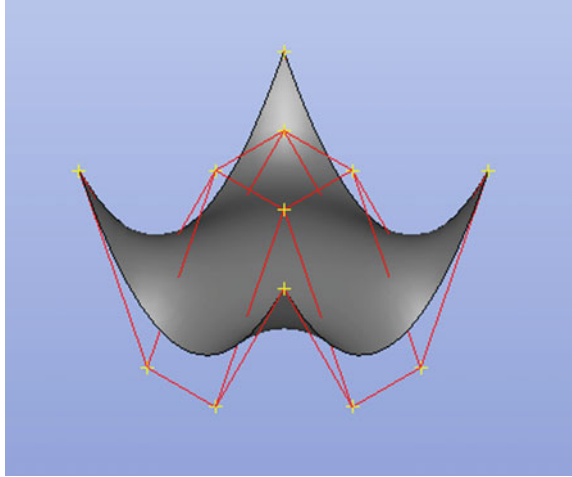
NURBS Curves and Surfaces

In this section the mathematical formulation of NURBS is reviewed. Initially, the creation of raw geometry is shown and then the final formulation of trimmed patches is analyzed (Fig. 1).

Mathematical Formulation

NURBS geometry is a generalization of B-spline geometry (Piegl and Tiller 1995). B-splines are the result of the combination of piecewise polynomial functions called

Fig. 1 A single NURBS patch along with its 4×4 control net



basis functions. Assuming the interpolation of n control values along a parametric direction u , the i -th basis function of degree p is defined as:

$$N_i^p(u) = \frac{u-U_i}{U_{i+p}-U_i} N_i^{p-1}(u) + \frac{U_{i+p+1}-u}{U_{i+p+1}-U_{i+1}} N_{i+1}^{p-1}(u)$$

$$N_i^0(u) = \begin{cases} 1 & \text{if } U_i \leq u < U_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $U_i - i \in [1, (n + p + 1)]$ – are the non-decreasing knot values used to segment the parameter space.

A NURBS curve of p degree and n control points \mathbf{P}_i and associated weights w_i is evaluated at parameter u as:

$$\mathbf{C}(u) = \frac{\sum_{i=1}^n N_i^p(u) \cdot w_i \mathbf{P}_i}{\sum_{k=1}^n N_k^p(u) \cdot w_k} \quad (2)$$

Similarly, a NURBS surface (generalization of a tensor product B-spline (Piegl and Tiller 1995)) of combined degrees $p \times q$, controlled by a $n \times m$ control grid of points and associated weights is evaluated at parameters u, v as:

$$\mathbf{S}(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m N_i^p(u) N_j^q(v) \cdot w_{i,j} \mathbf{P}_{i,j}}{\sum_{k=1}^n \sum_{l=1}^m N_k^p(u) N_l^q(v) \cdot w_{k,l}} \quad (3)$$

Trimmed NURBS Patches

Trimmed NURBS patches enable the representation of a much higher variety of shapes with less complex surfaces. The trimming procedure involves the creation of a closed wire of two-dimensional curves lying on the $u - v$ parametric space of the surface. The parametric space is then trimmed along this wire and the surface is not evaluated for u, v pairs outside the trimmed region. The number of curves, the wire consists of, is arbitrary and thus a multi-sided patch can be created using a single NURBS surface. It is useful if the two-dimensional curves, along with their three-dimensional curve-on-surface counterparts are NURBS curves themselves. A simple example of the trimming process is shown in Figs. 2.

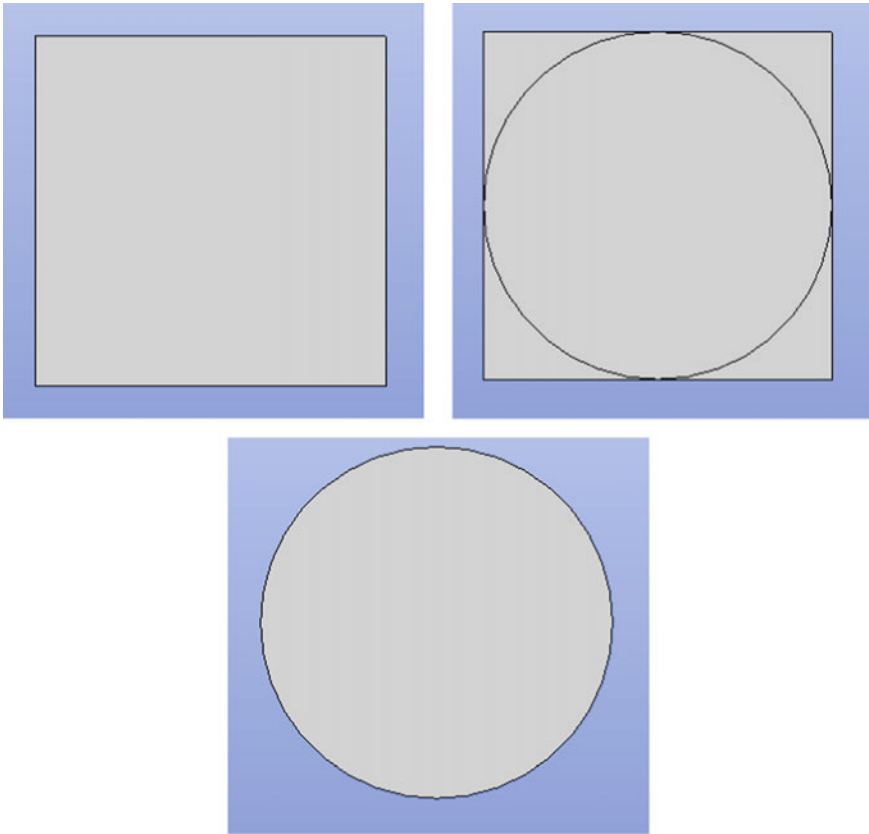


Fig. 2 Top left: A planar surface prior to the trimming process. Top right: The same surface with a trimming curve designed on it. Bottom: The circular disk which is a result of the trimming process

Adjoint Based Optimization and the Continuous Adjoint Technique

During recent years, in CFD gradient – based optimization, the Adjoint Technique (Pironneau 1974; Jameson 1988; Thvenin and Janiga 2008) has received much attention due to the fact that the cost of computing sensitivity derivatives of an objective function J is independent of the number of the design variables. Therefore, the Adjoint technique in its discrete (Giles et al. 2001; Vishnampet et al. 2015), or continuous (Jameson 1988; Othmer 2008, 2007; Giannakoglou and Papadimitriou 2008; Giannakoglou et al. 2015; Papoutsis-Kiachagias et al. 2014) form, is excellent for large scale optimization problems. In this article the continuous Adjoint formulation is used to calculate the sensitivity derivatives.

Primal Equations

The primal problem governed by the incompressible Reynolds - averaged Navier-Stokes equations can be written as:

$$R^p = -\frac{\partial u_i}{\partial x_i} = 0 \quad (4)$$

$$R_i^u = u_j \frac{\partial u_i}{\partial x_j} + \frac{\partial p}{\partial x_i} - \frac{\partial}{\partial x_j} \left[(v + \nu_t) \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] = 0 \quad (5)$$

In Eqs. 4 and 5 u denotes the components of the primal velocity and p is the primal pressure.

Adjoint Equations

Let J be a function to be minimized by the computation of an optimal set of design variables b_n , $n \in [1, N]$. The starting point of the continuous Adjoint formulation is the formulation of the augmented objective function J_{aug} . Assuming a computational domain Ω and its boundary S :

$$J_{aug} = J + \int_{\Omega} v_i R_i^u d\Omega + \int_{\Omega} q R^p d\Omega \quad (6)$$

In Eq. 6 v_i is the i – th component of the adjoint velocity and q is the adjoint pressure. It is obvious that since the primal state equations must hold, then $J_{aug} = J$. Minimization of J , therefore becomes minimization of J_{aug} .

$$\frac{\delta J_{aug}}{\delta b_n} = \frac{\delta J}{\delta b_n} + \int_{\Omega} v_i \frac{\partial R_i^u}{\partial b_n} d\Omega + \int_{\Omega} q \frac{\partial R^p}{\partial b_n} d\Omega + \int_S (v_i R_i^u + q R^p) n_k \frac{\delta x_k}{\delta b_n} dS \quad (7)$$

In Eq. 7, two differential operators can be seen: $\delta()/\delta b_n$ and $\partial()/\partial b_n$. The first operator denotes the total derivative and the second denotes the partial derivative. For a given quantity Φ , these operators are connected through

$$\frac{\delta \Phi}{\delta b_n} = \frac{\partial \Phi}{\partial b_n} + \frac{\partial \Phi}{\partial x_k} \frac{\delta x_k}{\delta b_n} \quad (8)$$

The field Adjoint equations are then formulated so as to make Eq. 7 independent of variations in the primal state variables. These are written as:

$$R^q = -\frac{\partial v_j}{\partial x_j} = 0 \quad (9)$$

$$R_i^v = v_j \frac{\partial u_j}{\partial x_i} + \frac{\partial q}{\partial x_i} - \frac{\partial(u_j v_i)}{\partial x_j} - \frac{\partial}{\partial x_j} \left[(v + v_i) \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right] = 0 \quad (10)$$

The time required to solve the Adjoint equations is equivalent to the time required to solve the primal problem. This makes apparent the strength of the Adjoint technique: the time required for the sensitivity calculation is two equivalent flow solutions regardless of the number of variables. The formula for the calculation of the sensitivity derivatives is omitted for the sake of space. The presentation of the adjoint formulation of turbulence models is omitted as well, for the sake of simplicity.

Geometry Morphing Method

During shape optimization based on the BRep of a model, an obvious challenge arises related to the continuity between the trimmed patches: Shape change involves the displacement of the control points of the underlying surfaces of the BRep patches. The fact that neighbor patches are seldom conforming untrimmed patches, makes the imposition of geometric continuity highly non-trivial, especially when trying to automate the procedure. Not many solutions have been proposed for this type of problem and the most promising was addressed by Xu et al. (2014).

Forming the Constraint Equations

The demand that two patches touch along a certain pathline that corresponds to two trimming curves, one on each patch, can be seen as the demand that they touch at an adequate number of points along the pathline. For a point with parameters (u, v) on patch 1 and (ξ, η) on patch 2, the constraint can be formulated as:

$$\mathbf{S}_1(u, v) = \mathbf{S}_2(\xi, \eta) \Leftrightarrow \mathbf{S}_1(u, v) - \mathbf{S}_2(\xi, \eta) = \mathbf{0} \tag{11}$$

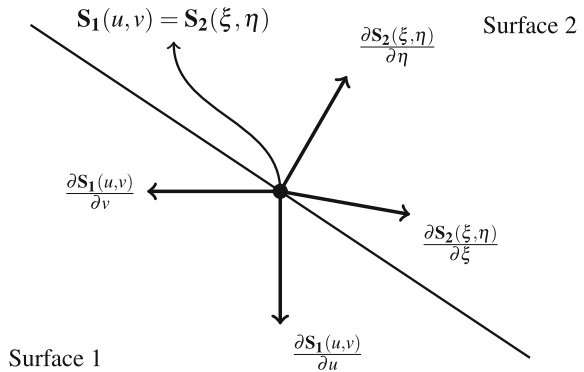
Imposing the constraint in Eq. 11 for a number of (u, v) and (ξ, η) pairs along the pathline, will make sure that the two patches touch at those points. If that number is big enough, then it can lead to the patches to fully touch along the pathline, practically ensuring C_0 continuity. C_0 continuity may not be enough when smoothness is required at the interface between the two patches. A way to impose C_1 continuity between the patches is to make sure that the vectors denoting the parametric derivatives of each surface are co-planar. Given three non co-linear vectors, a way to test co-planarity is to check if a vector can be written as a linear combination of the other two. If that condition holds, then the three vectors are co-planar. Therefore, in order to impose C_1 continuity between the two patches, one should formulate two necessary conditions (Fig. 3):

$$\frac{\partial \mathbf{S}_1(u, v)}{\partial u} = \alpha \cdot \frac{\partial \mathbf{S}_2(\xi, \eta)}{\partial \xi} + \beta \cdot \frac{\partial \mathbf{S}_2(\xi, \eta)}{\partial \eta} \tag{12}$$

$$\frac{\partial \mathbf{S}_1(u, v)}{\partial v} = \gamma \cdot \frac{\partial \mathbf{S}_2(\xi, \eta)}{\partial \xi} + \delta \cdot \frac{\partial \mathbf{S}_2(\xi, \eta)}{\partial \eta} \tag{13}$$

The coefficients $\alpha, \beta, \gamma, \delta$ are calculated for a pair of parametric coordinates by solving the equations:

Fig. 3 In this figure, a plain example of how the interface between two surfaces should look like. A point on the interface, should be evaluated using either surface mathematical definition. At the same time, the parametric derivatives of both surfaces on that point, have to be co-planar, if of course C_1 continuity is to be preserved



$$M \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{S}_1(u,v)}{\partial u} \cdot \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \xi} \\ \frac{\partial \mathbf{S}_1(u,v)}{\partial u} \cdot \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \eta} \end{bmatrix}$$

$$M \cdot \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{S}_1(u,v)}{\partial v} \cdot \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \xi} \\ \frac{\partial \mathbf{S}_1(u,v)}{\partial v} \cdot \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \eta} \end{bmatrix}$$

where

$$M = \begin{bmatrix} \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \xi} & \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \eta} & \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \xi} & \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \eta} \\ \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \eta} & \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \xi} & \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \eta} & \frac{\partial \mathbf{S}_2(\xi,\eta)}{\partial \xi} \end{bmatrix}$$

The Eqs. 11–13 are all depending linearly on control points of both surface patches. Thus they can be written in matrix form. If the first control net has dimensions $n_1 \times m_1$ and the second $n_2 \times m_2$ then the total number of control points is $N = n_1 \times m_1 + n_2 \times m_2$. A matrix $Q_{N \times 3}$ is created to store the control points of all the patches in ordered form. The first, second and third columns of Q store the x, y, z coordinates of the control points respectively. Using this matrix, the accumulation of all the constraints written in the form of Eqs. 11–13 can be written as:

$$A_{M \times N} \cdot Q_{N \times 3} = 0_{M \times 3} \quad (14)$$

or

$$A_{M \times N} \cdot \delta Q_{N \times 3} = 0_{M \times 3} \quad (15)$$

where A is the matrix containing the ordered linear coefficients apparent in all the constraint equations and M is the total number of constraint equations.

Null Space of the Coefficient Matrix

Clearly the trivial solution of Eq. 15 is of no interest. The set of non-trivial solutions to Eq. 15 or the Null Space (Meyer 2000) of matrix A is the primary focus. In order to evaluate the vectors that belong to the Null Space of A , eigen-vector/value analysis has to be done to the normalized constraint matrix $A^T A$. That is because $A^T A$ has the same Null Space as A but is much more well conditioned. Assuming an eigenvalue λ_i and the corresponding eigenvector \mathbf{u}_i of $A^T A$, it is obvious that:

$$\text{if } \lambda_i = 0 \text{ then } A^T A \mathbf{u}_i = 0$$

Thus the vector \mathbf{u}_i belongs to the Null Space. It is easy to prove two important properties of vectors belonging to the Null Space.

- Given a vector \mathbf{a} , that belongs to the Null Space, and a scalar k , then the vector $\mathbf{b} = k\mathbf{a}$ will also belong to the Null Space.

- Given two vectors \mathbf{a} , \mathbf{b} that belong to the Null Space, then the vector $\mathbf{c} = \mathbf{a} + \mathbf{b}$ will also belong to the Null Space.

A matrix with N columns and rank $r \leq N$ will have $N - r$ zero eigenvalues. Based on the above, one can show that any vector that can be written as:

$$\mathbf{x} = \sum_{i=1}^{N-r} k_i \cdot \mathbf{u}_i = [\mathbf{u}_1 \cdots \mathbf{u}_{N-r}] \cdot \begin{bmatrix} k_1 \\ \vdots \\ k_{N-r} \end{bmatrix} \quad (16)$$

will satisfy

$$A \cdot \mathbf{x} = 0$$

In Eq. 16 the matrix containing the eigenvectors belonging to the Null Space is called the Kernel of A or simply $Kernel(A)$. In the same equation, the coefficients k_i are arbitrary scalars (Null Space parameters). Based on these findings, one can say that any δQ calculated as:

$$\delta Q = Kernel(A) \cdot \delta K \quad (17)$$

will satisfy Eq. 15, for any $(N - r) \times 3$ matrix of arbitrary coefficients δK . The eigen-vectors/values of matrix $A^T A$ can be calculated through various orthogonal decompositions. In this work, the QR -decomposition (Golub and Van Loan 1996) is chosen because of the much better performance in sparse matrices. In case of a larger BRep model that consists of more than two surface patches, more constraint equations like Eq. 11–13 will have to be satisfied since continuity will have to hold at every interface between patches. At the same time the control points of additional patches will be stored in the matrix δQ , so even for larger models, Eq. 15 can be formulated accordingly. The analysis shown so far assumes that matrix δQ contains the entirety of the control points of each patch. That in practice is needless, because not all the control points of the BRep need to be constrained in order to impose the continuity. As a matter of fact, locality is one of the most useful properties of NURBS geometries. Therefore, the points along a trimming edge will be influenced by some of the control points of a patch. In practice, the matrix δQ contains only those constrained control points.

Calculation of the Shape Derivatives

Generally in NURBS-based optimization, the shape derivatives of an objective function are the derivatives with respect to the control points. Here, a similar procedure is followed. For the non-constrained control points, the process will be just that. However, the constrained control point derivatives will be used to calculate the derivatives of the Null Space parameters. That can be done by using a 3-step chain-rule. Assume

a CFD mesh and its boundary on the surface of the whole BRep model. If X is the point field of the boundary mesh with range $[1, nP]$, then:

$$X = [\mathbf{X}_1 \ \mathbf{X}_2 \ \cdots \ \mathbf{X}_{nP}] \text{ with } \mathbf{X}_i = (x_i, y_i, z_i) \quad (18)$$

For an objective function J , the sensitivity vectors can be calculated using an Adjoint CFD solver:

$$\frac{dJ}{dX} = \left[\frac{dJ}{d\mathbf{X}_1} \ \frac{dJ}{d\mathbf{X}_2} \ \cdots \ \frac{dJ}{d\mathbf{X}_{nP}} \right] \quad (19)$$

A boundary mesh point \mathbf{X}_i will belong to one of the BRep surfaces denoted by \mathbf{S}_i . Point inversion on that surface will yield the parameters (u_i, v_i) , such that:

$$\mathbf{S}_i(u_i, v_i) = \mathbf{X}_i \quad (20)$$

According to the locality property of NURBS, the point \mathbf{X}_i will have non-zero control point derivatives for some of the control points of surface \mathbf{S}_i . These according to Eq. 3, will be equal to the rational coefficients of the control points. For a control point $\mathbf{P}_{ip,jp}$ of surface \mathbf{S}_i :

$$\frac{d\mathbf{X}_i}{d\mathbf{P}_{ip,jp}} = \frac{N_{ip}^p(u_i)N_{jp}^q(v_i) \cdot w_{ip,jp}}{\sum_{k=1}^n \sum_{l=1}^m N_k^p(u_i)N_l^q(v_i) \cdot w_{k,l}} = R_{ip,jp}$$

If P denotes the non-constrained control points and R is the matrix with the coefficients $R_{ip,jp}$, then:

$$\begin{bmatrix} dJ/dP \\ dJ/dQ \end{bmatrix} = \left(\frac{dJ}{dX} \cdot R \right)^T \quad (21)$$

Based on Eq. 17, one can calculate the derivatives of constrained control points with respect to the Null Space parameters. Obviously, because of the linear dependence of the former on the latter, one can calculate:

$$\frac{dQ}{dK} = \text{Kernel}(A) \quad (22)$$

Therefore:

$$\frac{dJ}{dK} = \frac{dJ}{dQ} \frac{dQ}{dK} = \frac{dJ}{dQ} \cdot \text{Kernel}(A) \quad (23)$$

After the derivatives of Eqs. 23 and 21 are calculated, an optimizer can be used to provide a correction δK and δP ; δK is then applied to Eq. 17 in order to calculate a suitable constrained control point update δQ . Since δQ is calculated using Eq. 17, it will satisfy the constraint Eq. 15.

Optimization Algorithm

The analysis described previously, needs to be done once during the import of the BRep model. The constraints are linear and therefore, no re-calculation of the constraint matrix is required. Since the constraint matrix is constant, so is its Kernel. Briefly the optimization algorithm is listed below.

1. Import BRep model and optionally apply shape – fix algorithms.
2. Calculate the constraint equation matrix by applying C_0 and optionally C_1 continuity between patch interfaces. At the same time identify which control points are constrained.
3. Calculate and store the Kernel of the constraint matrix using the method shown in Sect. “Null Space of the Coefficient Matrix”.
4. Calculate the parameters of the boundary mesh points on the BRep using point inversion techniques (Ma and Hewitt 2003). Use these parameters to calculate the shape derivative matrix R .
5. Solve the Primal and the Adjoint flow problems and extract the sensitivities.
6. Use the sensitivities to calculate the unconstrained control point and Null Space parameter derivatives. Using these derivatives update the shape.
7. Update the mesh using the boundary mesh displacements provided by the updated BRep and check for stopping conditions. Either stop algorithm or go to step 5.

Results

The Geometry morphing method is coupled with the continuous Adjoint technique and is tested in two cases.

The S-Bend Climate Duct

The S-Bend duct (Fig. 4) is an air duct test case provided by VolksWagen AG. The goal of this test case is to minimize the power (pressure) losses of the flow subject to the deformation of the S-section of the duct. Of course, up to C_1 continuity is to be maintained at the interfaces between moveable patches and between the moving and the non-moving part as well. The geometry is rather challenging as from the 46 NURBS trimmed patches, the 28 belong to the S-section. These 28 patches have various complexities and the total number of control points on the S-section is 5830. Due to the high degree of the patches almost every control point is constrained through the Kernel of the constraint matrix. The flow analysis is done using a mesh of 700 K cells. The flow is laminar with an inlet velocity of $0.1m/s$ ($Re = 400$). A drop of 9.1% is achieved after 40 iterations. The convergence history can be seen in Fig. 6 and the final shape of the S-section in Fig. 5.

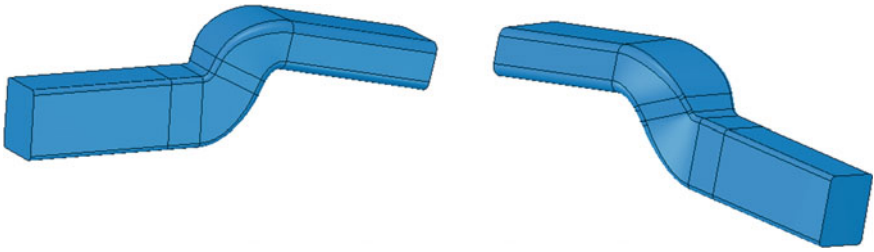


Fig. 4 Initial geometry of the S-Bend (Courtesy of Volkswagen AG)

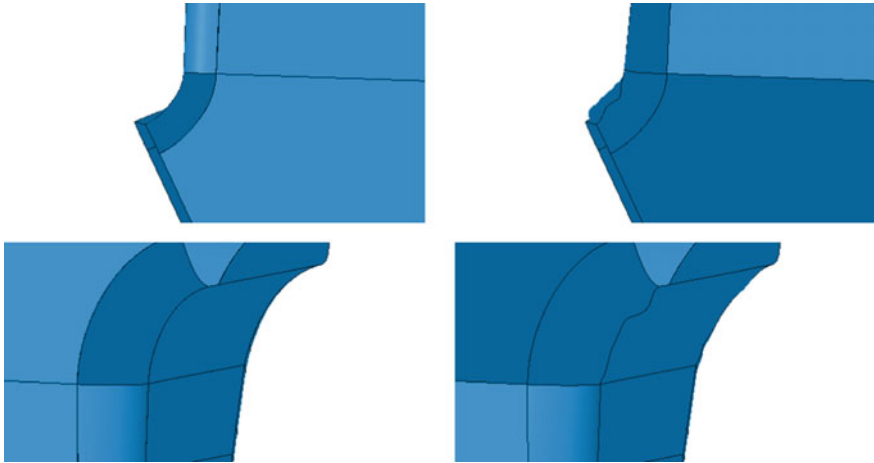
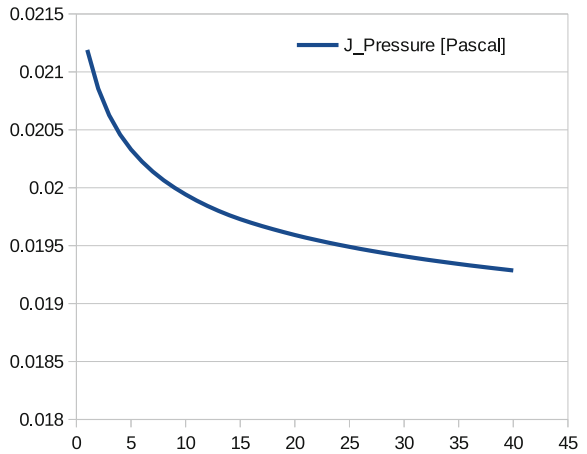


Fig. 5 Left: The initial CAD model at the regions with concentrated sensitivities. Right: The updated CAD model at the regions with concentrated sensitivities

Fig. 6 The convergence history of the objective function after 40 optimization cycles. The drop accounts for 9.1% the initial objective function value



The Drivaer Model

The Drivaer model is a reference vehicle model designed by TU Munich. The goal of this test case is to minimize the drag force on the car body. The analysis is done using a mesh of 15M cells. The velocity of the moving vehicle is set at 38.89 m/s (=140 Km/h), which yields $Re = 4.2E+6$. The flow is turbulent and thus the Primal problem is solved using the Spalart-Allmaras turbulence model. The flow is treated as steady and therefore averaging over time steps is required to calculate the flow variables and the sensitivities. Similarly to the S-Bend case, only a part of the vehicle is allowed to move. In particular, the regions towards the rear notchback area (trunk, spoiler, sides) (Figs. 7 and 8). The whole CAD model consists of 1300 BRep patches. The moveable part consists of 24 patches with a total of 5094 control points. C_0 and C_1 continuity is imposed in this case as well, constraining a total of 3094 control points. A drop in the objective function of 0.93% is noticed after five optimization cycles. The history of the unsteady objective function along with the mean values can be seen in Fig. 9.

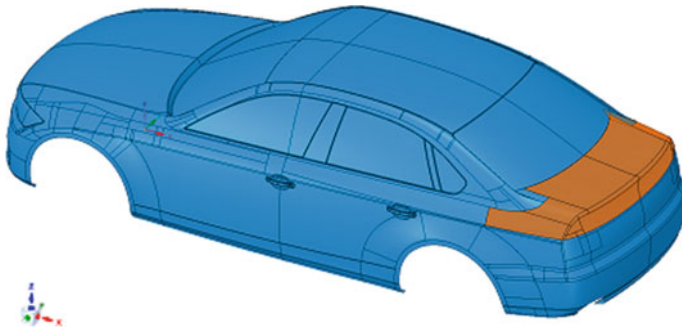


Fig. 7 Initial geometry of the Drivaer vehicle. In orange shading is the moveable part of the geometry for this case

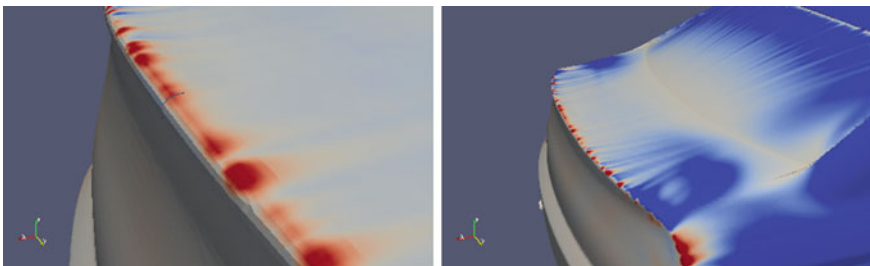


Fig. 8 In the left figure, the spoiler region of the Drivaer is demonstrated. On the right side of the vehicle the mesh before the update is shown. On the left side, the updated can be seen. In the right figure, the same region can be seen from farther. The coloring on the right and the left side of the vehicle matches the sensitivity map before and after the update respectively

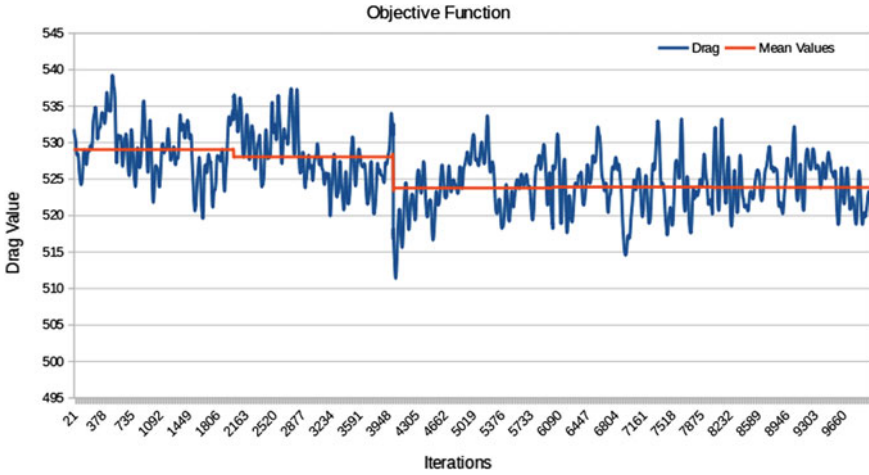


Fig. 9 In this figure the changes in the drag history after 5 updates can be seen. 2000 averaging steps were done per iteration. With blue color, the exact history can be seen. With orange color, the averaged objective values are shown. The change to the average values over the last 3 iterations is minimal

Conclusion

The Geometry Morphing method creates a CAD based framework for shape optimization that is independent of the CAD package used to design a model. Input to the method is a standard CAD file such as STEP or IGES. The BRep model stored in files of such types is described by NURBS parametric elements. Therefore, the shape optimization is performed by displacing the NURBS control points. The displaced shape is always constrained to be described by the same NURBS patches as the initial. Any design intent stored in the mathematical description of the NURBS, is passed to the optimized model. Continuity is ensured during optimization by imposing linear constraints with respect to control points at an adequate number of points on trimming curve regions. The constraints are maintained by making sure that the control point displacements always belong to the Null Space of the constraint matrix. The method is fast and accurate as (a) the time for the method to be initialized (constraint matrix, Null Space, point inversion) never exceeded 30 seconds and shape update was almost instantaneous and (b) continuity was always maintained at the level of the imported CAD tolerances. Furthermore, the method is fully automated to the extent that the BRep quality permits. The updated CAD model can always be exported to a standard file, for a post processing step by any modern CAD package. While Geometry Morphing is a reliable package independent CAD-related optimization method, it must be noted that in the end, optimization is done via the NURBS representation. NURBS are free form surfaces and there are always challenges when dealing with such geometric elements. For example, concentrated sensitivities can lead surface regions to morph into a high curvature state. Moreover, it would even be

possible to have a richer design space in the BRep than the mesh. That is a situation where the total number of control points of a moveable region of a model exceeds the number of the computational mesh points. This results in a wrinkly BRep surface as the sensitivity information is not evenly distributed along the control points.

Acknowledgements The work shown in this article is part of the IODA (Industrial Optimal Design using Adjoint CFD) Project. Research topic: Intuitive interfaces for optimisation parameterisation, constraint definition and automated mesh-to-CAD conversion. The project leading to this application, has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 642959.

References

- Giannakoglou KC, Papadimitriou DI (2008) Adjoint methods for shape optimization. Springer, Berlin, Heidelberg, pp 79–108
- Giannakoglou KC, Papadimitriou DI, Papoutsis-Kiachagias EM, Kavvadias IS (2015) Aerodynamic shape optimization using "Turbulent" adjoint and robust design in fluid mechanics. Springer, Cham, pp 289–309
- Giles MB, Duta MC, Miller JD, Pierce NA (2001) Algorithm developments for discrete adjoint methods
- Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. Johns Hopkins University Press, Baltimore, MD, USA
- Jameson A (1988) Aerodynamic design via control theory. *J. Sci Comput* 3(3):233–260
- Ma YL (2003) Hewitt WT (2003) Point inversion and projection for nurbs curve and surface: Control polygon approach. *Comput Aided Geom Des* 20(2):79–99
- Mantyla M (1988) Introduction to solid modeling. W. H. Freeman & Co., New York, NY, USA
- Meyer CD (ed) (2000) Matrix analysis and applied linear algebra. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA
- Nowacki H, Dannenberg L (1986) Approximation methods used in the exchange of geometric information via the VDA/VDMA surface interface. Springer, Berlin Heidelberg, pp 150–159
- Othmer C (2007) Implementation of a continuous adjoint for topology optimization of ducted flows (2007)
- Othmer C (2008) A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. *J Numer Methods Fluids* 58(8):861–877
- Papoutsis-Kiachagias, E., Kyriacou, S., and Giannakoglou, K. The continuous adjoint method for the design of hydraulic turbomachines. *Comput Methods Appl Mech Eng* 278(Complete):621–639 (2014)
- Piegl L, Tiller W (1995) The NURBS Book. Springer, London, UK
- Pironneau O (1974) On optimum design in fluid mechanics. *J Fluid Mech* 64(1):97–110
- Stroud I (2006) Boundary representation modelling techniques. Springer, New York, Secaucus, NJ, USA
- Thvenin D, Janiga G (2008) Optimization and computational fluid dynamics, 1st edn. Springer, Incorporated
- Vishnampet R, Bodony DJ Freund, JB (2015) A practical discrete-adjoint method for high-fidelity compressible turbulence simulations. *J Comput Phys* 285(C):173–192 (2015)
- Xu S, Jahn W, Miller J-D (2014) Cad-based shape optimisation with CFD using a discrete adjoint. *Int J Numer Methods Fluids* 74(3):153–168

CAD and Adjoint Based Multipoint Optimization of an Axial Turbine Profile



Ismael Sanchez Torreguitart, Tom Verstraete and Lasse Mueller

Abstract A computer-aided design (CAD) and adjoint based multipoint optimization of the LS89 high pressure axial turbine vane is presented. The aim is to reduce the entropy generation at both subsonic and transonic flow conditions by means of employing CAD and adjoint based methods during the optimization process. The performance metrics at design and off-design conditions are grouped into a single objective function using equal weights. A steady state Reynolds-Averaged density based Navier-Stokes solver and the one-equation transport Spalart-Allmaras turbulence model are used to predict the losses. The entropy generation is reduced whilst keeping the trailing edge thickness and the axial chord length as manufacturing constraints and the exit flow angle as a flow constraint, which is enforced via the penalty formulation. The resulting unconstrained optimization problem is solved by a L-BFGS-B algorithm. At every optimization iteration a new profile is constructed using B-splines and the grid is rebuilt by elliptic grid generation. The gradients used for the optimization are obtained via a novel approach in which both the CAD kernel and grid generation are differentiated using Algorithmic Differentiation techniques. The sensitivities of the objective function with respect to the grid coordinates are computed by a hand-derived adjoint solver. The off-design performance of the LS89 is significantly improved and the optimal geometry is analyzed in more detail.

I. Sanchez Torreguitart (✉) · L. Mueller
von Karman Institute for Fluid Dynamics (VKI), Chaussée de Waterloo 72, 1640
Rhode-Saint-Genèse, Belgium
e-mail: ismael.sanchez.torreguitart@vki.ac.be

L. Mueller
e-mail: lasse.mueller@vki.ac.be

T. Verstraete
Queen Mary University of London (QMUL), Mile End Road, London E1 4NS, UK
e-mail: t.verstraete@qmul.ac.uk

Nomenclature

c	chord length
c_{ax}	axial chord length
g	pitch
J_1	term proportional to entropy generation
J_2	exit flow angle
J_{MP}	multi point pseudo cost function
\dot{m}	mass flow
M_{ise}	isentropic Mach number
$M_{ise,2}$	downstream isentropic Mach number
P_{01}	inlet total pressure
P_{02}	outlet (downstream) total pressure
p_2	outlet (downstream) static pressure
R_{LE}	leading edge radius
R_{TE}	trailing edge radius
t	throat height
$t_{PS}^1, \dots, t_{PS}^4$	PS thickness
$t_{SS}^1, \dots, t_{SS}^9$	SS thickness
\mathbf{X}	grid x, y, z coordinates

Greek symbols

α	design vector
β_{in}	inlet angle
β_{out}	outlet angle
γ	stagger angle
$\frac{dJ}{d\alpha}$	performance sensitivity vector
$\frac{dJ}{d\mathbf{X}}$	adjoint sensitivity vector
$\frac{d\mathbf{X}}{d\alpha}$	grid sensitivity vector
φ_{PS}	pressure side trailing edge wedge angle
φ_{SS}	suction side trailing edge wedge angle
σ	solidity

Introduction

The pursuit of low-cost, efficient and accurate computational methods for numerical shape optimization in aerodynamics is critical for the aerospace industry. It enables the designer to make significant design improvements at the early stage of the design

chain before freezing the geometry and handing it over to manufacturing. One distinguishes between two main categories: local and global optimization algorithms. The computational resources for the global optimization algorithms can grow exponentially with the number of design variables, and hence their use in industry chain design is sometimes restricted to small test cases and/or preliminary design. The computational effort can be significantly reduced by using local optimization algorithms, such as gradient based methods, provided that the gradients are computed with the adjoint method, for which the cost is nearly independent of the number of design variables (Peter and Dwight 2010).

Choosing a suitable parameterization which is used for industrial design processes can reduce the post-processing efforts to deliver a feasible optimal shape for manufacturing. The parameterization of the geometry can be done at either the grid level or within the computer-aided design (CAD) environment. In the former case, a common approach is to use the grid point coordinates as design variables (Jameson 2004), which offers a very rich design space but the connection to the CAD geometry is lost. Since CAD is the industry adopted standard for the design of components, an additional step is required to transform the optimal shape defined by grid points back to smooth CAD shape, which can take significant time and it is not guaranteed that the final approximated CAD shape will meet the design requirements and constraints (Braibant and Fleury 1984; Samareh 2001). Also, the use of a smoother is mandatory to avoid the unbounded growth of high frequency oscillations, leading to irregular shapes. This inherently reduces the rich design space, yet still allows for the exploration of unconventional designs. To avoid previously mentioned problems, in this work we propose to keep the CAD geometry in the optimization loop. The use of CAD-based parameters such as stagger angle, leading and trailing edge radius, etc., however, requires an additional sensitivity computation step. The partial derivative of the grid coordinates with respect to the design variables, also referred in this work as the grid sensitivities, need to be computed as well. This could be done by using finite differences, but the accuracy of the gradients is dependent upon the chosen step-size for each design variable. This work aims to circumvent the accuracy issues due to limited arithmetic precision or truncation errors of finite differences by using Algorithmic Differentiation (AD) (Griewank and Walther 2008) for the CAD kernel and the grid generation (Sanchez Torreguitart et al. 2016).

The well-known LS89 (Arts et al. 1990) high pressure axial turbine nozzle guide vane was originally designed and optimized at the Von Karman Institute for Fluid Dynamics for a subsonic isentropic outlet Mach number of 0.9, by an inverse method (Van den Braembussche et al. 1990) based on Euler and potential flow analysis solvers, using the difference between the calculated velocity distribution and the required one to modify the profile geometry. A CAD and adjoint based approach was used in Montanelli et al. (2015) to perform a single and multi-point optimization of the LS89 to reduce the total pressure losses by constraining the outlet mass flow whilst keeping the leading and trailing edge geometries and the profile thicknesses fixed. Recently, substantial aerodynamic improvements were achieved at design point (Sanchez Torreguitart et al. 2017) by means of using a different parameterization and possibly richer design space. As shown in Sanchez Torreguitart et al. (2017), a

16 % reduction in total pressure loss was achieved whilst keeping the axial chord length, trailing edge radius and exit flow angle fixed. However, the performance of the optimal profile is deemed to deteriorate significantly at off-design conditions. Given that a turbomachine typically operates over a range of different aerodynamic conditions, this paper aims at optimizing the performance of the LS89 axial turbine profile at off-design conditions. The design optimization is carried out by using a CAD and adjoint based approach, in which a combination of forward AD and hand-derived reverse differentiation is being used for the grid generation and the flow solver respectively.

Methodology

Figure 1 shows a schematic view of the optimization process. In this section, some of the individual components of the optimization flow chart will be discussed in more detail.

In order to create the geometry in an automated fashion it is important to have a robust parameterization. The turbine profile parameterization shown in Figs. 2a, b is based on the description given in Pierret (1999).

Various engineering and CAD-based design parameters that are relevant to the aerodynamic performance (e.g., solidity, stagger angle, etc.) and to the manufacturing requirements (e.g., axial chord length, trailing edge radius) are used to define the profile. First, a camber line is constructed with a 2nd order Bézier curve defining its control points (P_{LE} , P_{mid} , P_{TE}), as shown in Fig. 2b. The suction side (SS) and pressure side (PS) curves are constructed as B-spline curves by defining the position of the control points relative to the camber line. The profile is closed by circular arcs at the leading edge (LE) and trailing edge (TE). Equal curvature geometric continuity (i.e., second order derivative continuity) is maintained between the SS and PS B-splines at the leading edge by applying the appropriate normal distance of the first control point relative to the camber line for this purpose. A total number of 22 design variables is used for the optimization process. The axial chord length and the trailing edge radius are kept fixed as manufacturing constraints.

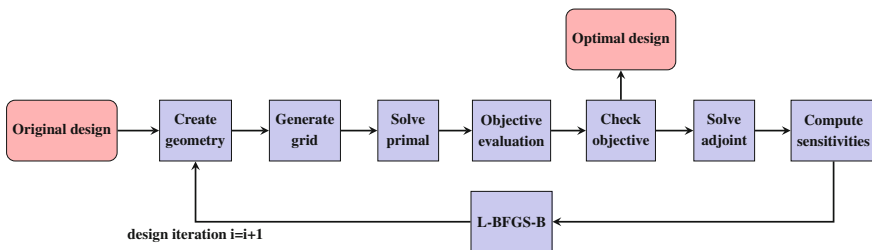


Fig. 1 Optimization flow chart

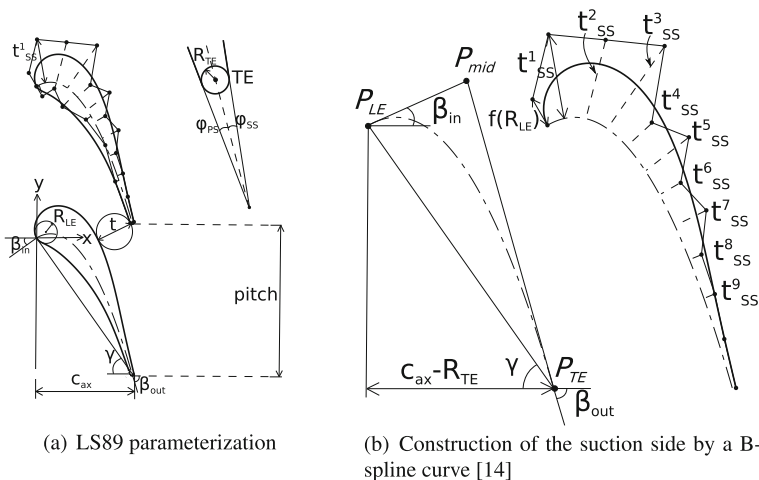


Fig. 2 CAD-based parameterization of the LS89

After creating the profile geometry, a multi-block structured grid is rebuilt for every optimization iteration. A mesh-independence study was carried out in order to find the appropriate mesh settings, which were kept the same during the optimization process. Authors refer to Sanchez Torreguitart et al. (2016) for details about the grid generation.

The flow solver employs a cell-centered finite volume discretization on multi-block structured grids. The three dimensional compressible RANS equations are solved with an adaptation of the JT-KIRK scheme described in Xu et al. (2015), which is an implicit Runge-Kutta time integration scheme accelerated by local time-stepping and multigrid. The fluid is considered to be a calorically perfect gas and the eddy-viscosity hypothesis is used to account for the effect of turbulence. Convective fluxes are computed using the second order accurate Roe’s approximate Riemann solver (Roe 1981) with a MUSCL-type reconstruction (Van Leer 1979). Viscous fluxes are calculated with a central discretization scheme. The numerical dissipation of the scheme is controlled by the entropy correction by Harten and Hyman Harten (1983). Oscillations near shocks are suppressed by a van-Albada type limiter (Venkatakrishnan 1993). Boundary conditions are imposed weakly by utilizing the dummy cell concept (Blazek 2001). The negative Spalart-Allmaras turbulence model (Allmaras and Johnson 2012) is used for the turbulence closure problem assuming fully turbulent flow from the inlet ($Re_{inlet} \approx 2 \times 10^5$).

After solving the primal solver, a performance metric that is proportional to the entropy generation J_1 and the exit flow angle J_2 at the outlet of the domain can be computed for each operating condition with expressions 1 and 2 respectively.

$$J_1 = \frac{\int_{out} p \rho^{1-\gamma} V_x dy}{\dot{m}_{out}} \tag{1}$$

$$J_2 = \text{atan} \left(\frac{\overline{V}_y}{\overline{V}_x} \right) \quad (2)$$

After non-dimensionalizing them with $J_{1,ref}$ and $J_{2,ref}$ they are combined into a pseudo cost function via the penalty term method (expression 3), which transforms the constrained problem in an unconstrained optimization formulation.

$$J_{MP} = \sum_{op=1}^3 \frac{1}{3} \left(\frac{J_{1,op}}{J_{1,ref}} \right) + \omega \left(\frac{J_{2,op3}}{J_{2,ref}} - 1 \right)^2 \quad (3)$$

The left term of the pseudo cost function is a weighted average of the non-dimensional entropy generation, which is computed with equal weights for the following operating points (op): $M_{ise,2} = 0.9$ (op = 1), $M_{ise,2} = 0.955$ (op = 2) and $M_{ise,2} = 1.01$ (op = 3). By grouping the performance metrics at design and off-design conditions into one single objective function, the multipoint objective optimization can be treated as a single objective one. The right term of expression 3 is the penalty term, which becomes larger the more the exit flow angle deviates from the target value. In order to reduce the computational effort, the penalty term is computed only for the 1.01 Mach transonic operating point, which is the most challenging operating point when it comes to satisfy the aerodynamic constraint. We hereby make the assumption that the flow turning at $M_{ise,2} = 1.01$ is always smaller than at $M_{ise,2} = 0.9$, which we later validate on the baseline and optimized profiles. The ω is the penalty coefficient and was selected by trial and error.

After evaluating the cost function, it is necessary to compute the gradients. The adjoint solver computes the gradients of the cost function J (e.g., entropy generation or exit flow angle) with respect to the grid point coordinates \mathbf{X} (i.e., $dJ/d\mathbf{X}$). Similarly to the flow solver, the adjoint solver uses the same stabilization JT-KIRK scheme of Xu et al. (2015). The hand derived discrete adjoint solver assumes that the eddy viscosity does not change with geometry variations (frozen turbulence), which is a valid approach for most engineering design applications.

Next, the performance sensitivities ($dJ/d\boldsymbol{\alpha}$) for each cost function J with respect to the design variables $\boldsymbol{\alpha}$ are computed by a scalar product of the adjoint-based sensitivities ($dJ/d\mathbf{X}$) with the grid sensitivities ($d\mathbf{X}/d\boldsymbol{\alpha}$) as follows:

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{dJ}{d\mathbf{X}} \frac{d\mathbf{X}}{d\boldsymbol{\alpha}}. \quad (4)$$

In this work, the AD tool ADOL-C is used to compute the grid sensitivities $d\mathbf{X}/d\boldsymbol{\alpha}$ in one single evaluation of the primal at a relatively low cost by using the forward vector mode approach. The $dJ/d\boldsymbol{\alpha}$ gradients were also computed by finite differences and compared against the ones obtained with the method described in this study, showing a reasonably good agreement (Figs. 3a, b).

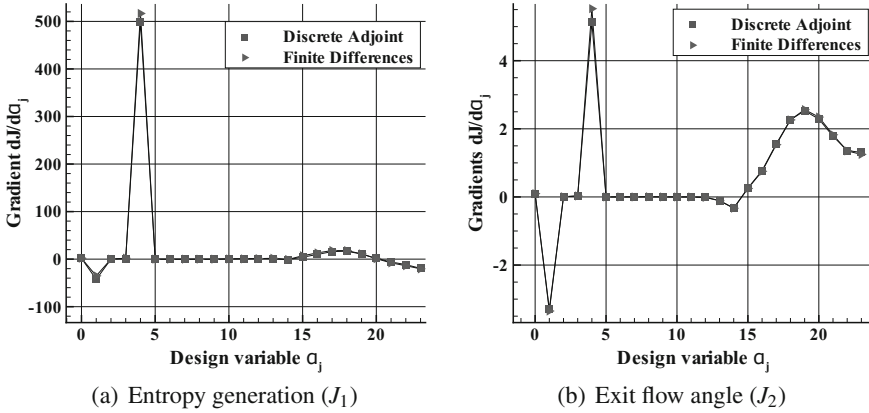


Fig. 3 Adjoint based gradients compared to finite difference approximations

Finally, the gradients are given to the Quasi-Newton L-BFGS-B algorithm (Zhu et al. 1997), which is available in the python SciPy package (Jones et al. 2017), and it is used to find the new design vector.

Results

The optimizer performed a total number of 30 iterations, from which 12 of them were line search iterations in order to find an appropriate step size. Figure 4 shows the evolution of the non-dimensional cost functions during the optimization process after excluding the line search iterations. The solid line represents the evolution of the first term of the pseudo cost function (expression 3), which has been reduced by 6.4% whilst satisfying the exit flow angle aerodynamic constraint (i.e., $J_{2,op3} / J_{2,ref} \geq 1.0$), which is indicated by the line with discrete points. The evolution of the non-dimensional entropy generation for each operating point is also shown on Figure 4. The optimizer was able to improve the aerodynamic performance of the profile for each operating point, but the largest improvements were made for the operating point with a downstream isentropic Mach number of 1.01.

Figure 5 compares the baseline and the optimal profiles and Table 1 summarizes the main geometrical changes.

Isentropic Mach Number Comparison

Figures 6, 7 and 8 compare the isentropic Mach number distribution of the baseline and the optimal profiles for a downstream isentropic Mach number of 0.9, 0.955 and

Fig. 4 Cost functions evolution

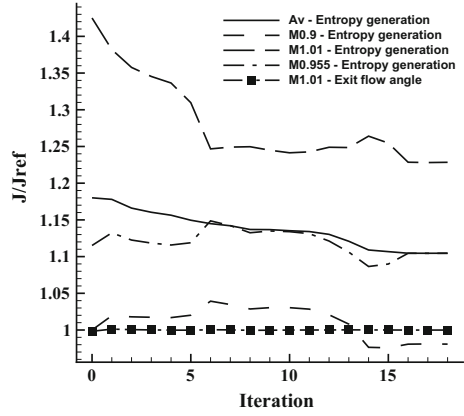


Fig. 5 Geometry comparison of the baseline and optimal profiles

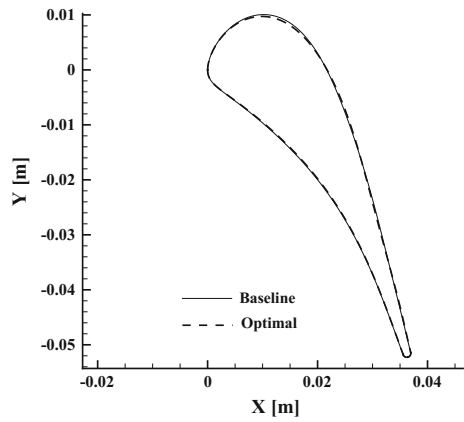


Table 1 Comparison of the main geometrical changes between the baseline and optimal profiles

	Acronyms	Units	Baseline	Optimal	Variation
Pressure side trailing edge wedge angle	φ_{PS}	(deg)	2.500	2.498	-0.0022°
Suction side trailing edge wedge angle	φ_{SS}	(deg)	4.000	4.002	0.002°
Chord	c	(mm)	64.310	64.254	-0.09%
Stagger angle	γ	(deg)	54.925	54.890	-0.035°
Inlet metal angle	β_{in}	(deg)	0.0	0.0032	0.0032°
Outlet metal angle	β_{out}	(deg)	74.000	73.981	-0.0190°
Leading edge radius	R_{LE}	(mm)	4.126	4.097	-0.71%
Solidity	σ	(-)	1.118	1.114	-0.40%
Pitch	g	(mm)	57.500	57.683	0.32%
Pitch/chord	g/c	(-)	0.894	0.898	0.41%

Fig. 6 Isentropic Mach number comparison for a downstream $M_{ise} = 0.9$

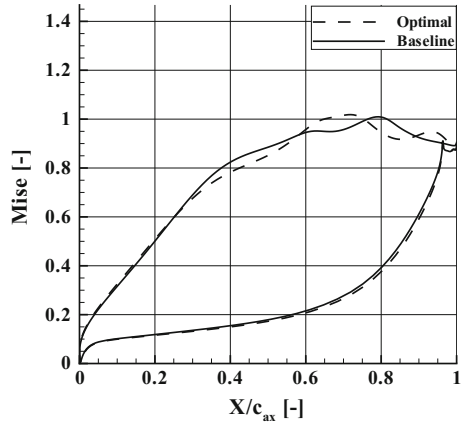


Fig. 7 Isentropic Mach number comparison for a downstream $M_{ise} = 0.955$

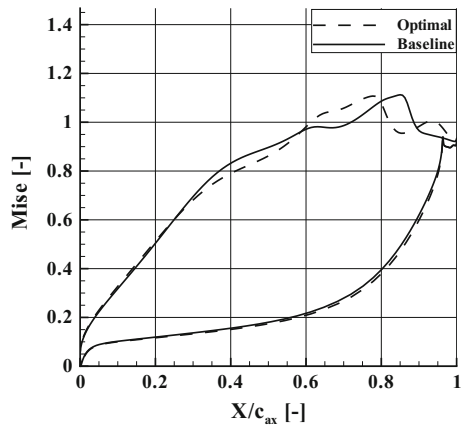
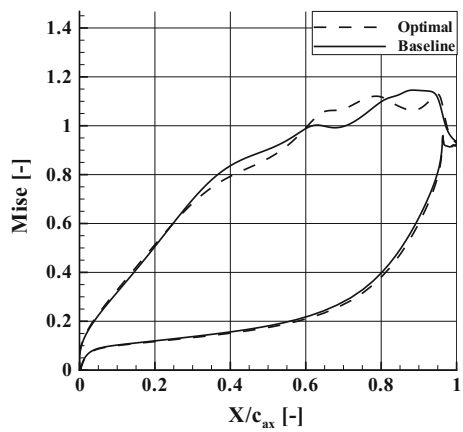


Fig. 8 Isentropic Mach number comparison for a downstream $M_{ise} = 1.01$



1.01 respectively. Figure 8 shows that a shock is located approximately at $X/c_{ax} = 0.95$ for the baseline. The optimizer was able to reduce slightly the suction side peak Mach number from 1.145 to 1.129 and moved the peak Mach number location from $X/c_{ax} = 0.9$ to $X/c_{ax} = 0.95$.

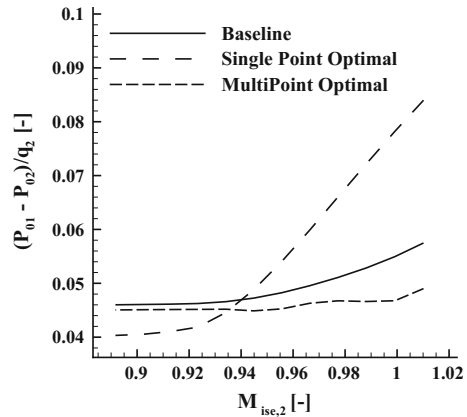
Off-Design Performance

Table 2 shows the main changes in entropy generation, exit flow angle, mass flow, and total pressure losses ($P_{01} - P_{02}$), for the three selected operating points. The largest aerodynamic improvements were achieved for the transonic operating point where the total pressure loss reduction was of the order of 14.06 %. The mass flow changes are kept within a reasonable limit and the exit flow angle is in all the three operating points above the baseline value. Figure 9 shows the total pressure loss coefficient at off-design conditions for the baseline, single point (Sanchez Torreguitart et al. 2017) and multipoint optimal profiles, which is defined as the total pressure difference between the inlet and outlet divided with the dynamic head at the outlet plane. The single point optimal profile is the same geometry that was investigated in more detail in Sanchez Torreguitart et al. (2017), which targets only low losses the $M_{ise,2} = 0.9$ operating point, and hence does not perform as good on other operating points. At off-design conditions, the aerodynamic improvements of the single point optimal profile are reduced as the downstream isentropic Mach number is increased. Beyond $M_{ise,2} = 0.94$, the performance deteriorates rapidly and the baseline would have lower total pressure losses than the single point optimal profile. In contrast, the multipoint optimal profile has lower total pressure losses than the baseline at off-design conditions for the whole Mach number range analysed in the present study. However, below $M_{ise,2} = 0.935$ the single point optimal profile would yield lower total pressure losses.

Table 2 Changes in performance at the different operating points

Operating point	$M_{ise,2} = 0.9$ (%)	$M_{ise,2} = 0.955$ (%)	$M_{ise,2} = 1.01$ (%)
Entropy generation variation	-1.92	-0.96	-13.79
Exit flow angle variation	0.05	0.16	0.17
Mass flow variation	0.12	-0.36	-0.05
Total pressure losses variation	-1.98	-5.94	-14.06

Fig. 9 Variation of the total pressure loss in function of the downstream isentropic Mach number Variation of the total pressure loss in function of the downstream isentropic Mach number



Conclusions

This paper presents a multipoint optimization of the LS89 axial turbine vane profile for a downstream isentropic Mach number of 0.9 (design point), 0.955 and 1.01. The off-design performance of the LS89 was significantly improved. The largest aerodynamic improvements were achieved at the transonic operating point where the total pressure losses were reduced by 14% whilst keeping the exit flow angle fixed. By keeping the CAD representation in the optimization loop, it is not necessary to convert the optimal grid back to a smooth CAD shape. As shown in this work it is possible to maintain manufacturing constraints fixed, like the axial chord length and the trailing edge radius. The successful application of the CAD and adjoint based methods presented in this study for a 2D profile will proof its merits in future 3D test cases with richer design spaces.

References

- Allmaras SR, Johnson FT, Spalart PR (2012) Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In: ICCFD7-1902, 7th international conference on computational fluid dynamics
- Arts T, Lambert De Rouvroit M, Rutherford A (1990) Aero-thermal investigation of a highly loaded transonic linear turbine guide vane cascade. A test case for inviscid and viscous flow computations. NASA STI/Recon technical report N 91, 23437
- Blazek J (2001) Computational fluid dynamics: principles and applications, 2nd edn. Elsevier Science Ltd, Amsterdam
- Braibant V, Fleury C (1984) Shape optimal design using B-splines. Comput Methods Appl Mech Eng 44(3):247267
- Griewank A, Walther A (2008) Evaluating: principles and techniques of algorithmic differentiation. Siam

- Harten A, Hyman JM (1983) Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. *J Comput Phys* 50(2):235269
- Jameson A (2004) Efficient aerodynamic shape optimization. AIAA paper 4369:2004
- Jones E, Oliphant T, Peterson P et al (2001) SciPy: open source scientific tools for Python. <https://www.scipy.org/>. Accessed 8 Feb 2017
- Montanelli H, Montagnac M, Gallard F (2015) Gradient span analysis method: application to the multipoint aerodynamic shape optimization of a turbine cascade. *J. Turbomach* 137(9):091006
- Peter JE, Dwight RP (2010) Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches. *Comput Fluids* 39(3):373391
- Pierret S (1999) Designing turbomachinery blades by means of the function approximation concept based on artificial Neural Networks, Genetic Algorithms, and the Navier-Stokes equations. PhD thesis, Von Karman Institute for Fluid Dynamics
- Roe PL (1981) Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys* 43:357372
- Samareh JA (2001) Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA J* 39(5):877884
- Sanchez Torreguitart I, Verstraete T, Mueller L (2016) CAD kernel and grid generation algorithmic differentiation for turbomachinery adjoint optimization. In: 7th European congress on computational methods in applied sciences and engineering, Hersonissos, Crete, Greece, June
- Sanchez Torreguitart I, Verstraete T, Mueller L (2017) Optimization of the LS89 axial turbine profile using a cad and adjoint based approach. In: Proceedings of 12th European conference on turbomachinery fluid dynamics & thermodynamics, ETC12, Stockholm, Sweden
- Steger J, Sorenson R (1979) Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *J Comput Phys* 33(3):405–410
- Thompson JF, Soni BK, Weatherill NP (1998) Handbook of grid generation. CRC Press, Boca Raton
- Van den Braembussche R, Leonard O, Nekkrouche L (1990) Subsonic and transonic blade design by means of analysis codes. *Computational methods for aerodynamic design (inverse) and optimization*, AGARD CP, p 463
- Van Leer B (1979) Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J Comput Phys* 32(1):101–136
- Venkatakrisnan V (1993) On the accuracy of limiters and convergence to steady state solutions. AIAA Paper 93-0880, Jan 1993
- Xu S, Radford D, Meyer M, Müller J-D (2015) Stabilisation of discrete steady adjoint solvers. *J Comput Phys* 299:175195
- Zhu C, Byrd RH, Lu P, Nocedal J (1997) Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans Math Softw (TOMS)* 23(4):550560

A Comparative Study of Two Different CAD-Based Mesh Deformation Methods for Structural Shape Optimization



Marc Schwalbach, Tom Verstraete, Jens-Dominik Müller
and Nicolas Gauger

Abstract This work introduces and compares two different CAD-based mesh deformation methods. The methods are used within an adjoint structural shape optimization, which is part of an evolving CAD-based adjoint multidisciplinary optimization framework for turbomachinery components. During an optimization, the CAD geometry is updated at each design iteration, such that the structural mesh has to be deformed appropriately. The mesh is deformed in three stages. First, the nodes along the edges of the outer mesh are displaced to match the shape of the CAD edges, which are given by B-spline curves. Next, the remaining outer mesh nodes are displaced to match the shape of the CAD faces, which are given by B-spline surfaces. Finally, the outer mesh node deformations are used to solve for the inner node deformations using either an inverse distance interpolation or the linear elasticity analogy. Coupling the mesh deformation with an adjoint structural solver enables gradient computations of structural constraints with respect to CAD design parameters. To compare the robustness of the two mesh deformation methods, a CAD-based structural shape optimization using each method was performed.

M. Schwalbach (✉)
von Karman Institute for Fluid Dynamics,
Waterloosesteenweg 72, 1640 Sint-Genesius-Rode, Belgium
e-mail: marc.schwalbach@vki.ac.be

T. Verstraete · J.-D. Müller
Queen Mary University of London, Mile End Road, London E1 4NS, UK
e-mail: t.verstraete@qmul.ac.uk

J.-D. Müller
e-mail: j.mueller@qmul.ac.uk

N. Gauger
TU Kaiserslautern,
Paul-Ehrlich-Strasse 34, 67663 Kaiserslautern, Germany
e-mail: nicolas.gauger@scicomp.uni-kl.de

Nomenclature

$m \in \mathbb{N}$	number of FEM mesh nodes
$m_i \in \mathbb{N}$	number of inner FEM mesh nodes
$m_o \in \mathbb{N}$	number of outer FEM mesh nodes
$n \in \mathbb{N}$	number of CAD design parameters
$u, v \in \mathbb{R}$	B-Spline foot points
$u_B \in \mathbb{R}$	foot point of begin vertex
$u_B^M \in \mathbb{R}$	morphed foot point of begin vertex
$u_E \in \mathbb{R}$	foot point of end vertex
$u_E^M \in \mathbb{R}$	morphed foot point of end vertex
$\mathbf{b} \in \mathbb{R}^{3m}$	load vector
$\mathbf{u} \in \mathbb{R}^{3m}$	FEM mesh displacements
$\mathbf{u}_{inner} \in \mathbb{R}^{3m_i}$	inner FEM mesh displacements
$\mathbf{u}_{outer} \in \mathbb{R}^{3m_o}$	outer FEM mesh displacements
$\mathbf{x} \in \mathbb{R}^{3m}$	FEM mesh coordinates
$\bar{\mathbf{x}} \in \mathbb{R}^{3m}$	adjoint FEM mesh coordinates
$A \in \mathbb{R}^{3m \times 3m}$	stiffness matrix
$C \in \mathbb{R}^3$	B-spline curve
$C^M \in \mathbb{R}^3$	morphed B-spline curve
$E \in \mathbb{R}$	Young's modulus
$P \in \mathbb{R}^3$	mesh point
$P^M \in \mathbb{R}^3$	morphed mesh point
$S \in \mathbb{R}^3$	B-spline surface
$S^M \in \mathbb{R}^3$	morphed B-spline surface
$V_B \in \mathbb{R}^3$	begin vertex
$V_B^M \in \mathbb{R}^3$	morphed begin vertex
$V_E \in \mathbb{R}^3$	end vertex
$V_E^M \in \mathbb{R}^3$	morphed end vertex
$\nu \in \mathbb{R}$	Poisson's ratio
$\sigma_{max} \in \mathbb{R}$	maximum von Mises stress
$\bar{\sigma}_{max} \in \mathbb{R}$	adjoint maximum von Mises stress
$\boldsymbol{\alpha} \in \mathbb{R}^n$	CAD design parameters
$\bar{\boldsymbol{\alpha}} \in \mathbb{R}^n$	adjoint CAD design parameters
$\Delta \in \mathbb{R}$	steepest descent step size

Introduction

Multidisciplinary optimizations (MDO's) have been extensively used to optimize turbomachinery components using gradient-free methods (Mueller et al. 2013; Verstraete 2008). Gradient-free methods are non-intrusive and do not require source

code access to implement. On the other hand, they require a high number of iterations to converge and are limited by the curse of dimensionality. This means that the computational effort grows exponentially with the number of design parameters. Gradient-based methods use gradients to converge towards a local optimum using less iterations, but require the gradient of the cost function with respect to design parameters.

Computing the gradient using a non-invasive approach, such as second order finite differencing (FD), comes at a high computational cost that is proportional to the number of design parameters. Adjoint methods (Pironneau 1974; Jameson 1988) allow the computation of gradients at a cost proportional to the number of costs and constraints, which is typically far less than the number of design parameters. Algorithmic differentiation (AD) (Griewank and Walther 2008; Naumann 2012) can be used to derive a discrete adjoint model from source code.

State of the art adjoint optimizations in turbomachinery focus on aerodynamic cost functions and constraints (Willeke and Verstraete 2015; Luo et al. 2014; Walther and Nadarajah 2013). Structural constraints are of high importance because the resulting shape should not only be aerodynamically optimal, but also structurally feasible. As a result, an adjoint structural solver has to be integrated within an MDO framework. In this work, the von Karman Institute's optimization framework CADO (Verstraete 2010) is used. A CAD-based optimization is strived for, which would also allow for geometrical constraints to be imposed. For example, one may want to impose a constraint on the shape's curvature for manufacturing purposes. The sensitivities of structural constraints with respect to mesh node coordinates can be computed within this framework (Schwalbach et al. 2016), but the sensitivities with respect to CAD design parameters are missing to perform a CAD-based optimization.

An essential component required for CAD-based gradient optimization methods is a mesh morphing tool which adapts the mesh to the modifications of the CAD model, while maintaining the same mesh connectivity and node count. A re-meshing strategy would alter the mesh topology, causing the objective function to become discontinuous.

Two CAD-based mesh deformation methods are presented in Section “[Mesh Deformation Method](#)”. Algorithmically differentiated versions of the mesh deformations can be used to perform an adjoint shape optimization, which will be discussed in Section “[CAD-Based Structural Shape Optimization](#)”. The optimization results are used to compare the robustness of the two methods in Section “[Optimization Results and Comparison](#)”.

Mesh Deformation Method

While iterating through a CAD-based optimization process, the CAD design parameters are updated, morphing the CAD geometry. Based on the updated geometry, the structural mesh should be deformed to compute the cost function and the sensitivities of the updated design. Since the CAD geometry defines the outer skin of

the structural mesh, it is used to compute an accurate deformation of the outer mesh nodes.

The CAD-based mesh deformation algorithm can be broken down into three hierarchical steps:

1. morph edge nodes
2. morph face nodes
3. morph inner nodes

The mesh deformations can be expressed as displacements $\mathbf{u} \in \mathbb{R}^{3m}$, where m is the number of structural mesh nodes. The first two steps are identical in both methods and are used to compute the outer mesh node displacements $\mathbf{u}_{outer} \in \mathbb{R}^{3m_o}$, where m_o denotes the number of FEM mesh nodes on the external surface of the mesh. The mesh deformation methods differ in the last step of morphing the inner mesh nodes. The three steps are briefly outlined in this section.

Morph Edge Nodes

First, the edges of the mesh are morphed based on the deformation of the CAD geometry edges, which are represented as B-spline curves $C(u)$. The displacements of the first and last vertex nodes of the mesh edge are identical to the first and last points of the B-spline curve. Using this constraint, the displacements of the points along the curve can be solved for. Solving for the displaced foot points u in parametric CAD space, rather than fitting the mesh coordinates \mathbf{x} to the B-spline curve $C(u)$, reduces the degrees of freedom to one. The constraint of requiring these mesh points to be on the edge is then implicitly applied.

To visualize this procedure, consider the example presented in Fig. 1. After the CAD geometry is updated, the B-spline curve C , which describes the edge, turns into the curve C^M . As a result, the begin and end vertices (V_B and V_E) are morphed into V_B^M and V_E^M . A mesh node P now has to be morphed into P^M by morphing its parametric coordinate u into u^M . This is done using the parametric coordinates

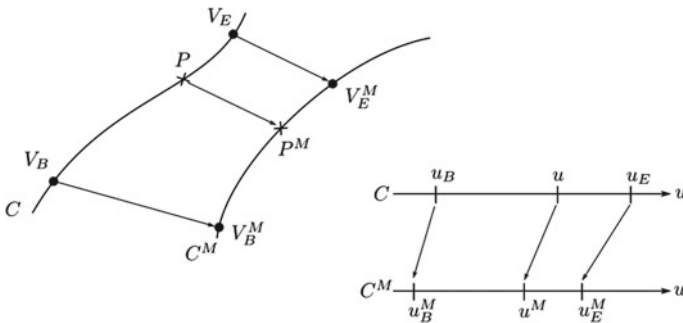


Fig. 1 Morphing of edge mesh nodes using parametric CAD space

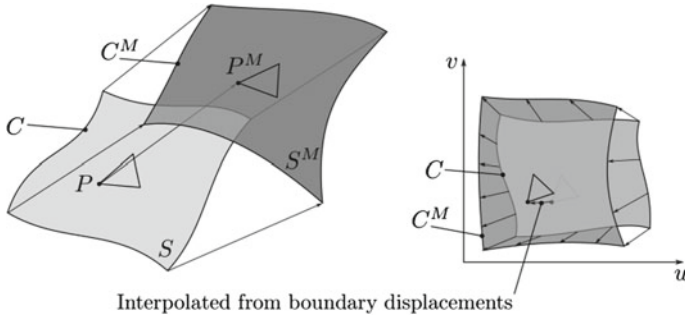


Fig. 2 Morphing of face mesh nodes using parametric CAD space

u_B, u_E of the begin and end vertices before morphing and their morphed parametric coordinates u_B^M, u_E^M :

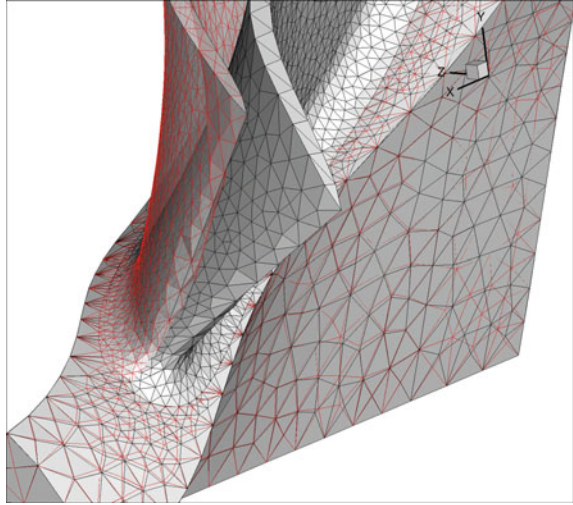
$$u^M = u_B^M + \frac{u_E^M - u_B^M}{u_E - u_B} (u - u_B) \tag{1}$$

A linear spring analogy is used to relax the points along the curve. Performing this step for each edge of the CAD faces results in the displacements of the structural mesh edge nodes.

Morph Face Nodes

After having displaced the mesh nodes along the edges, the next step is to displace the remaining outer mesh nodes according to the CAD faces. Each CAD face is represented by a B-spline surface $S(u, v)$, which morphs into S^M . Using the computed edges from step 1 as boundary conditions, the inner (u, v) foot points of the displaced CAD face are computed using an inverse distance interpolation. As with step 1, the displaced nodes are solved in parametric (u, v) space to reduce the degrees of freedom to two, which automatically satisfies the constraint that the displaced mesh nodes have to remain on the CAD face. An illustration of this procedure is shown in Fig. 2. This is done for each face of the geometry.

Fig. 3 Structural mesh after deformation using the inverse distance method



Morph Inner Nodes

Method 1: Using the Inverse Distance Interpolation

The first mesh deformation method computes the inner node displacements $\mathbf{u}_{inner} \in \mathbb{R}^{3m_i}$ using an inverse distance interpolation, where m_i denotes the number of inner structural mesh nodes (Verstraete 2017). The inverse distance interpolation is based on the displacements of the outer nodes $\mathbf{u}_{outer} \in \mathbb{R}^{m_o}$, i.e. the skin of the structural mesh, which are determined by the first two steps. An example of the resulting deformed mesh is shown in Fig. 3.

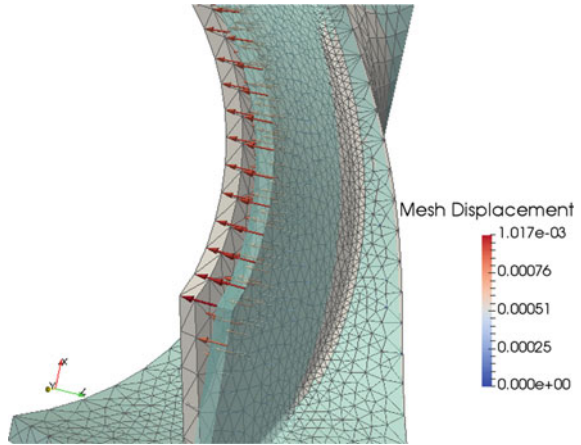
Method 2: Using the Linear Elasticity Analogy

The second mesh deformation method uses a linear elasticity analogy to solve for the inner node displacements $\mathbf{u}_{inner} \in \mathbb{R}^{3m_i}$. The outer node displacements \mathbf{u}_{outer} are used as boundary conditions to the linear elastic problem

$$A\mathbf{u} = \mathbf{b}, \quad (2)$$

where A is the stiffness matrix and \mathbf{b} is the load vector. A structural solver based on the finite element method (FEM) is used to solve for the mesh displacements \mathbf{u} . A visualization of the resulting mesh deformation is presented in Fig. 4. For now, global material properties are used for the entire mesh, i.e. the Young's modulus E and Poisson's ratio ν are constant throughout. These properties could also be defined

Fig. 4 Structural mesh after deformation using the linear elasticity analogy



locally to adjust the stiffness of certain parts of the mesh. Furthermore, the adjoint CSM solver can be recycled for the adjoint implementation of the mesh deformation.

CAD-Based Structural Shape Optimization

Previous work within this framework has enabled the computation of structural sensitivities with respect to FEM mesh nodes (Schwalbach et al. 2016). The adjoint structural solver was differentiated using CoDiPack (Albring et al. 2015). While these gradients could be used to perform node-based optimizations, an optimization using CAD design parameters is aspired (Schwalbach and Verstraete 2016). There are several reasons that motivate this approach, one being that CAD design parameters provide a more intuitive design space for engineers compared to computational meshes. Additionally, important geometric constraints can be imposed directly on an optimization, e.g. constraining the blade's curvature for manufacturing purposes, minimum thickness requirements, etc. CAD-free parametrizations, such as free-form deformation, have a greater difficulty fulfilling such constraints. For gradient-based optimizations, the structural sensitivities with respect to the CAD parameters are required. This is achieved by closing the gap between the CAD-based mesh deformation and the structural solver.

For a structural optimization, a typical cost function would be the maximum von Mises stress $\sigma_{max} \in \mathbb{R}$. As a design space, consider the CAD parameters $\alpha \in \mathbb{R}^n$, which are used as inputs into the CAD kernel to generate the CAD geometry. In CADO, these could e.g. include the blade angle (Fig. 5) and thickness distributions (Fig. 6). Thus, for a gradient-based optimization, the gradients

$$\frac{\partial \sigma_{max}}{\partial \alpha} \in \mathbb{R}^n \quad (3)$$

Fig. 5 Blade angle distribution

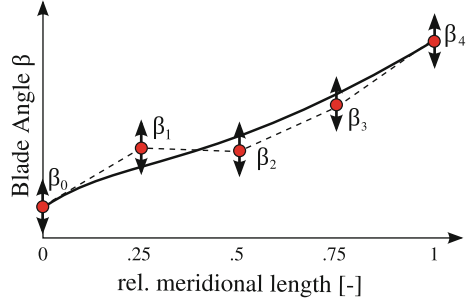
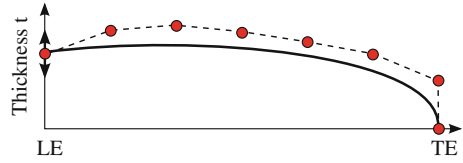


Fig. 6 Blade thickness distribution



are required. The respective adjoint model

$$\bar{\mathbf{x}} = \bar{\sigma}_{max} \frac{\partial \sigma_{max}}{\partial \mathbf{x}}, \tag{4}$$

can be used to compute the gradients by seeding the model with $\bar{\sigma}_{max} = 1$. Previous work has enabled the calculation of the gradient

$$\frac{\partial \sigma_{max}}{\partial \mathbf{x}} \in \mathbb{R}^m \tag{5}$$

with respect to the FEM mesh nodes $\mathbf{x} \in \mathbb{R}^m$. Knowing that the FEM mesh \mathbf{x} is dependent on the CAD geometry, which is generated based on the CAD parameters α , it can be established that

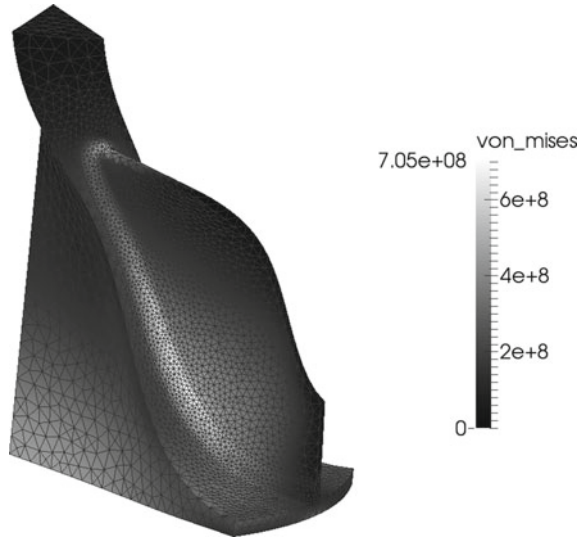
$$\frac{\partial \sigma_{max}}{\partial \alpha} = \frac{\partial \sigma_{max}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \alpha}. \tag{6}$$

The gradient (5) can be calculated using the adjoint structural solver, while the gradient $\frac{\partial \mathbf{x}}{\partial \alpha}$ can be computed by differentiating the mesh deformation in either forward or reverse AD. Using reverse AD, the structural sensitivities (5) could be used to seed the adjoint model of the mesh deformation

$$\bar{\alpha} = \frac{\partial \mathbf{x}^T}{\partial \alpha} \bar{\mathbf{x}} = \frac{\partial \mathbf{x}^T}{\partial \alpha} \frac{\partial \sigma_{max}}{\partial \mathbf{x}}, \tag{7}$$

computing the gradient (6) with a single adjoint evaluation.

Fig. 7 Mesh of initial radial turbine design



Optimization Results and Comparison

A radial turbine mesh, discretized using 10-node tetrahedral elements, was used to perform a structural optimization. The initial mesh is shown in Fig. 7, which contains approximately 85,000 nodes. The objective of the optimization was to minimize the maximum von Mises stress σ_{max} , which is approximated using the p -norm

$$\sigma_{max} = \left(\sum_{i=0}^{m-1} \sigma_i^p \right)^{\frac{1}{p}}, \tag{8}$$

using CAD parameters α as design variables. A steepest descent algorithm

$$\alpha^{i+1} = \alpha^i - \Delta \frac{\partial \sigma_{max}}{\partial \alpha} \tag{9}$$

with a constant step size of $\Delta = 10^{-8}$ was used. The criterion to remesh is if the value of the new cost function σ_{max}^{i+1} is greater than 5% of the current optimum σ_{max}^* or if the cost function reduction is less than 0.01%.

The resulting optimized geometries are shown in Figs. 8 and 9 for the linear elastic and inverse distance methods, respectively. Both deformation methods lead to similar geometries, reducing the von Mises stresses in the blade fillet area, by increasing the thickness of the back plate near the center and decreasing the thickness in the outer radii.

A convergence comparison of two optimizations using the different mesh deformation methods is shown in Fig. 10. Overall, using the linear elastic deformation

Fig. 8 Optimized geometry using linear elastic mesh deformation

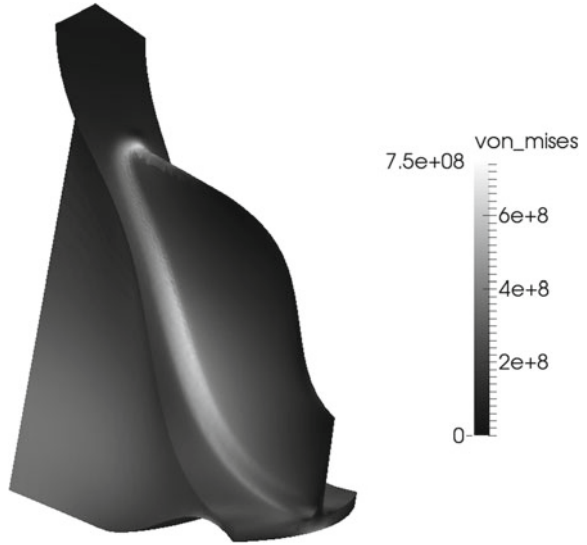
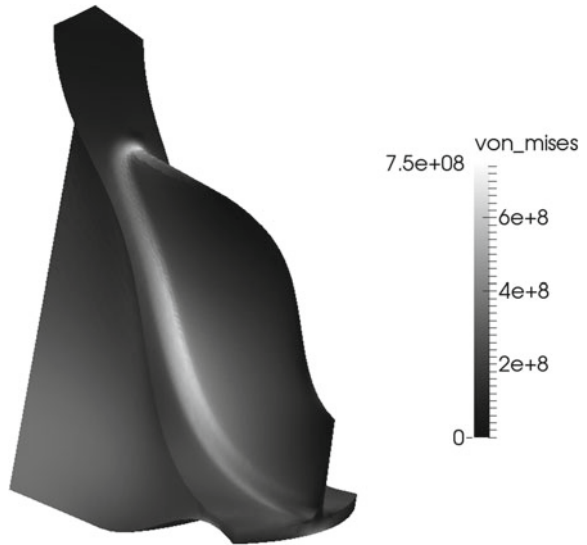


Fig. 9 Optimized geometry using inverse distance mesh deformation



method resulted in a greater cost function reduction of 10.4%, compared to the reduction of 9.1% with the inverse distance method. The linear elastic method requires its first remeshing at iteration 16, while the inverse distance method requires it at iteration 6. Both remeshing occurrences were triggered by the updated cost function being 5% greater than the current optimum.

The kink in the convergence curve (Fig. 10) of the linear elastic method at iteration 6–7 can be attributed to the deformation at the back plate as shown in Fig. 11a. The

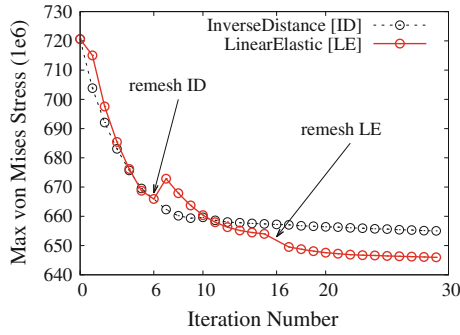


Fig. 10 Optimization convergence for inverse distance (ID) mesh deformation and linear elastic (LE) mesh deformation

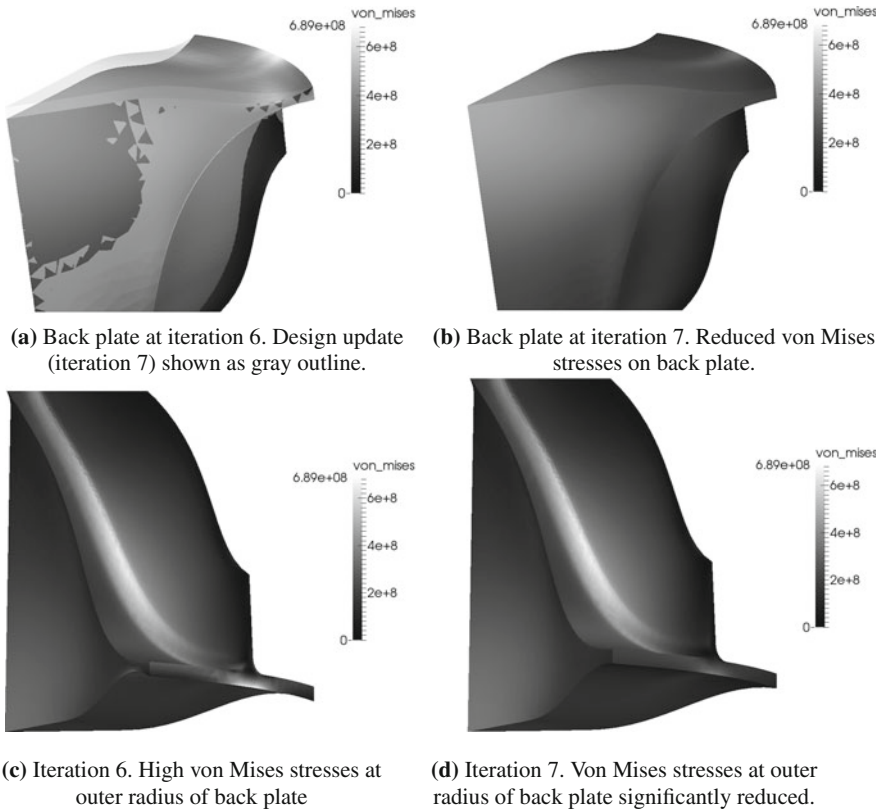


Fig. 11 Evolution of radial turbine geometry using linear elastic mesh deformation method

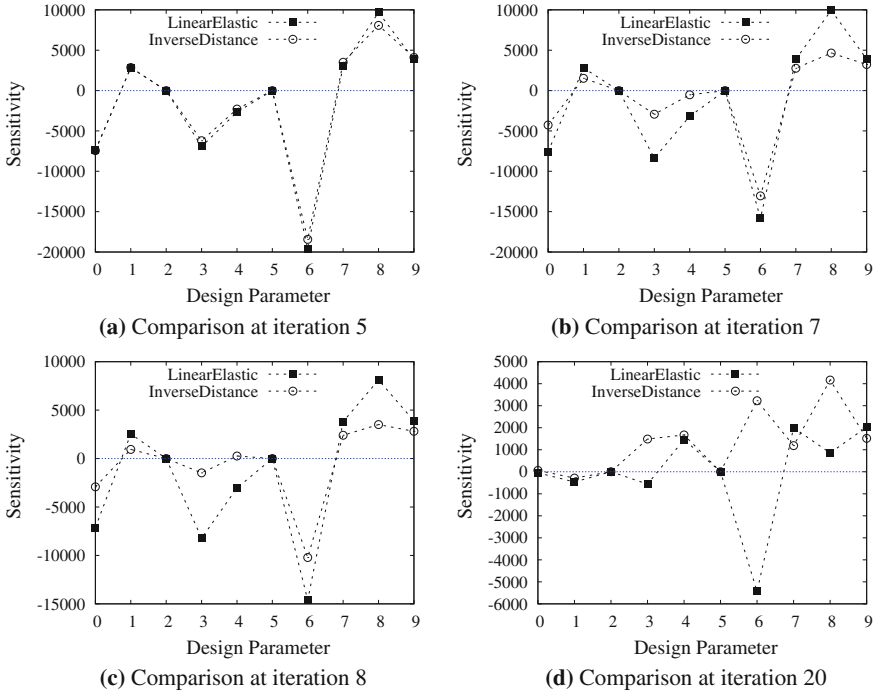


Fig. 12 Sensitivity comparisons between optimizations using the linear elastic and inverse distance mesh deformation methods at different optimization iterations

design update effectively reduces the high von Mises stresses at the outer radius of the back plate (Fig. 11b). As a result, one of the areas of high von Mises stresses is removed (Fig. 11c, d), affecting the maximum von Mises stress computed using the p -norm (8).

Why have the two optimizations converged to different designs? A comparison of the sensitivities between the two optimizations is shown in Fig. 12. In Fig. 12a, before any remeshing occurs, one can see that the sensitivities with respect to the design parameters match up well. At iteration 6, the first remesh is triggered for the optimizer using the inverse distance method. Hence, at iteration 7, the two optimizations are using different meshes. The resulting effect is reflected in the sensitivity discrepancies shown in Fig. 12b. The gradient with respect to design parameter 4 is much closer to zero. At iteration 8 (Fig. 12c), the sign of this gradient actually differs for the two optimizations, which affects the direction that the optimizer takes for this design parameter. The difference in sign is even more apparent at iteration 20 (Fig. 12d). Despite these discrepancies, the final geometries appear similar in shape.

Conclusion

This work introduced two unstructured mesh deformation methods for CAD-based adjoint optimizations. The mesh deformations take a CAD-based approach, especially for the deformation of the outer mesh nodes. This ensures an accurate conformity with the updated CAD geometry. The inner mesh deformations are computed using either an inverse distance interpolation or the linear elastic analogy with the help of a structural solver.

The structural solver, which has adjoint capabilities, additionally enables the computation of sensitivities of the structural cost function, e.g. the maximum von Mises stress, with respect to CAD design parameters. The sensitivities can be used to perform a structural shape optimization based on CAD design parameters. A structural optimization of a radial turbine has been performed using the different mesh deformation methods introduced in Section “[Mesh Deformation Method](#)”.

The comparison between the two methods, discussed in Section “[Optimization Results and Comparison](#)”, shows that remeshing can potentially lead to different optima. The inverse distance method triggers remeshing at an earlier stage compared to the linear elastic deformation. From this point on, the optimizers iterate towards different designs due to a sign difference in the sensitivities. As a result, using the linear elastic deformation, the optimizer has achieved a greater cost function reduction of 10.4%, compared to a 9.1% reduction using the inverse distance method. Both methods have converged towards similar shapes.

Future work would involve coupling the structural and fluid disciplines, as well as adding a vibration analysis. Specifically, coupling the adjoint chain of operations from CAD design parameters to structural constraints with an adjoint computational fluid dynamics (CFD) code. A CAD-based adjoint multidisciplinary optimization of a turbomachinery component can then be carried out.

Acknowledgements The work presented in this paper has received funding from the European Commission through the IODA project under grant agreement number 642959.

References

- Albring T, Sagebaum M, Gauger NR (2015) Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. In: 16th AIAA/ISSMO multidisciplinary analysis and optimization conference, p 3240
- Griewank A, Walther A (2008) Evaluating derivatives: principles and techniques of algorithmic differentiation. Siam
- Jameson A (1988) Aerodynamic design via control theory. *J Sci Comput* 3(3):233–260
- Luo J, Zhou C (2014) Multipoint design optimization of a transonic compressor blade by using an adjoint method. *J. Turbomach.* 136(5):051005
- Mueller L, Alsalihi Z, Verstraete T (2013) Multidisciplinary optimization of a turbocharger radial turbine. *J. Turbomach.* 135(2):021022
- Naumann U (2012) The art of differentiating computer programs: an introduction to algorithmic differentiation, vol 24. Siam

- Pironneau O (1974) On optimum design in fluid mechanics. *J. Fluid Mech.* 64(1):97–110
- Schwalbach M, Verstraete T (2016) Towards multidisciplinary adjoint optimization of turbomachinery components. In: Papadrakakis VPM, Papadopoulos V, Stefanou G (eds) *Proceedings of the VII European congress on computational methods in applied sciences and engineering*, pp 3999–4010
- Schwalbach M, Verstraete T, Gauger NR (2016) Developments of a discrete adjoint structural solver for shape and composite material optimization. In: *The 7th international conference on algorithmic differentiation*
- Verstraete T (2008) *Multidisciplinary turbomachinery component optimization considering performance, stress, and internal heat transfer*. University of Ghent, Ph.D. thesis
- Verstraete T (2010) *Cado: a computer aided design and optimization tool for turbomachinery applications*. In: *2nd international conference on engineering optimization*, Lisbon, Portugal, September, pp 6–9
- Verstraete T, Müller L, Müller JD (2017) CAD based adjoint optimization of the stresses in a radial turbine. In: *Proceedings of ASME Turbo Expo 2017: turbine technical conference and exposition*
- Walther B, Nadarajah S (2013) Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor. *J Turbomach* 135(2):021017
- Willeke S, Verstraete T (2015) Adjoint optimization of an internal cooling channel u-bend. In: *ASME Turbo Expo 2015: turbine technical conference and exposition*. American Society of Mechanical Engineers

Node-Based Adjoint Surface Optimization of U-Bend Duct for Pressure Loss Reduction



G. Alessi, L. Koloszar, Tom Verstraete and J. P. A. J. van Beeck

Abstract The pressure loss reduction inside the U-bends of internal cooling channels is of crucial importance to increase the performance of cooling systems of gas turbines. The optimization technique proposed in the present work is based on the continuous adjoint shape method and is implemented in the OpenFOAM open-source framework. The calculated gradients of the objective function are linked to a node-based constrained morphing routine, allowing the modification of the shape towards an optimum design with minimal pressure loss. The integration with a robust mesh morpher solver leads to successive automatic steps towards the design improvement. Design modifications take into account constraints and limitations related to the chosen design. The feasibility of the design is guaranteed by the application of a smoothing function with the aim to avoid rough external surfaces.

Introduction

Numerical tools often assist the design development. Thanks to the continuous growth of the computational resources and the increased accuracy of numerical simulation models, it is possible to compare different design prototypes and investigate the influence of several parameters with the aim to improve its performance. However,

G. Alessi (✉) · L. Koloszar · J. P. A. J. van Beeck
von Karman Institute for Fluid Dynamics, 1640 Sint-Genesius-Rode, Belgium
e-mail: giacomo.alessi@vki.ac.be

L. Koloszar
e-mail: lilla.koloszar@vki.ac.be

J. P. A. J. van Beeck
e-mail: vanbeeck@vki.ac.be

G. Alessi
Civil Engineering Department, Catholic University of Leuven, 3000 Leuven, Belgium

T. Verstraete
Queen Mary University of London, London E1 4NS, UK
e-mail: t.verstraete@qmul.ac.uk

due to the designs complexity, many different parameters are usually coupled and their effect on the design could be difficult to assess. As a consequence, in the past decades has been carried out a large development of computational routines that autonomously maximize the design performance, i.e. the so-called optimization algorithms. Nonetheless, many design optimization techniques suffer from dimensionality problems, since the numerical effort significantly increases with the degrees of freedom. Especially when costly CFD simulations are required to evaluate the performance, the search for optimal shapes can only be executed effectively with limited design freedom. Gradient based optimization techniques, however, still allow for rich design spaces if the adjoint method is used to compute the sensitivities (Pironneau 1974), since its computational cost is independent from the number of design variables. The direction of improvement can be obtained at a fixed cost independently of the size of the design variables space, considerably reducing the overall procedure time and the computational power needed. The adjoint method is increasingly becoming the tool of choice for gradient optimizations, whose application can be found in different fields (Jameson 1988; Othmer 2014; Goit and Meyers 2014).

An optimization routine based on the adjoint method has been developed in this work and has been tested with an internal aerodynamic application. The optimization test case is linked to turbomachinery industrial applications, in particular to the cooling system of gas turbines (Han et al. 2000). The U-bend shape is particularly representative of the cooling system design, being the major source of thermodynamic efficiency loss. Due to the large number of parameters that could influence this application, the adjoint method represents an efficient approach to the problem.

The U-Bend Test Case

Cooling systems of gas turbines, in most of the cases, use air bled from the high pressure compressor, leading to a penalty in thermodynamic efficiency. A design of the internal cooling passages for minimal pressure losses would improve the global efficiency of the gas turbine. The U-bends that connect consecutive passages are amongst the largest contributors to the pressure losses in the cooling system and deserve special attention during the design phase. Hence, its design has been considered separated from the rest of the internal cooling structure in this optimization work. The considered design and flow configuration has received particular attention both from research groups and industries, thus various approaches and optimization strategies have been applied to it (Verstraete and Li 2013; Zehner et al. 2009; Namgoong et al. 2008).

The numerical domain considered in the present work is represented by a circular U-bend of square section (hydraulic diameter $D_h = 0.075$ (m)), whose geometrical details can be found in Coletti et al. (2011). A velocity profile with a bulk velocity of $U_0 = 8.4$ (m/s) and a turbulence intensity of $T.I. = 5\%$ has been imposed at the inlet (Verstraete et al. 2011). A zero pressure boundary condition is applied on the outlet and a no-slip condition on the wall.

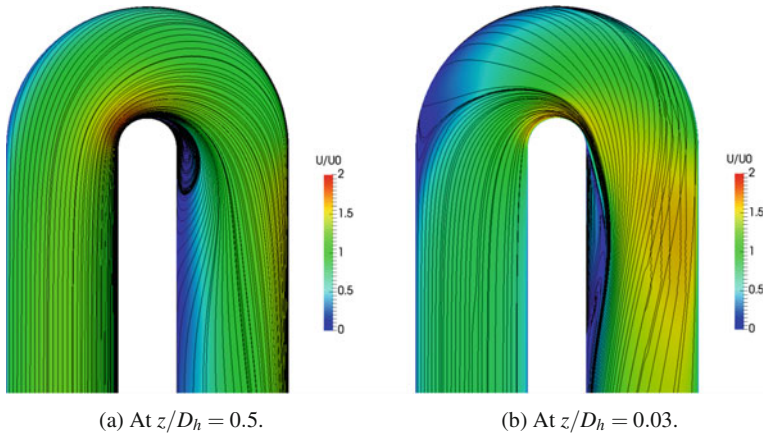


Fig. 1 Velocity field and streamlines in the initial 3D U-bend design

Reynolds-averaged Navier-Stokes simulations have been performed using a structured grid, assuring a maximum y_+ value of 0.94. The Launder-Sharma low-Reynolds $k - \varepsilon$ turbulence model has been used. The velocity field obtained and its streamlines are shown in Fig. 1, in particular: (a) at $z/D_h = 0.5$ and (b) at $z/D_h = 0.03$. The flow accelerates approaching the bend, it reaches the maximum velocity around the inner wall while it decelerates along the outer wall. At the symmetry plane, $z/D_h = 0.5$, the flow starts to separate before the end of the bend forming a separation bubble. A longer recirculation bubble is present at $z/D_h = 0.03$. The flow field characteristics obtained are in agreement with the numerical ones in Coletti et al. (2011).

The routine has been first tested with a 2D test case to verify the reliability of the procedure and after applied to the 3D case. The 2D geometry refers to the symmetry plane of the 3D U-bend design, thus the flow field characteristics are similar to it with the exception of the presence of a longer separation bubble than the one obtained in the 3D case at $z/D_h = 0.5$, as shown in Fig. 2.

The aim of the study is the minimization of the total pressure loss, thus the considered cost function J is represented by:

$$J = p_T^{in} - p_T^{out} = \left(p + \frac{1}{2} \rho U^2 \right)_{in} - \left(p + \frac{1}{2} \rho U^2 \right)_{out} \quad (1)$$

The Adjoint Method in the Context of Shape Optimization

In the adjoint formulation two systems of equations have to be solved, the Navier-Stokes equations and the adjoint equations. The first step of the optimization process is to solve the Navier-Stokes equations (primal system) to obtain converged flow variables, as shown in Section “The U-Bend Test Case”; thereafter the convergence

Fig. 2 Velocity field and streamlines in the initial 2D U-bend design

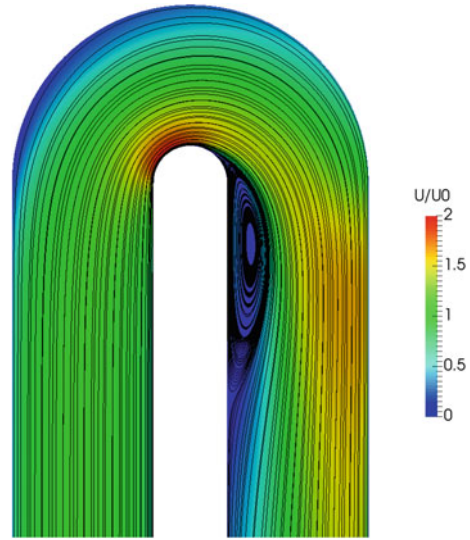
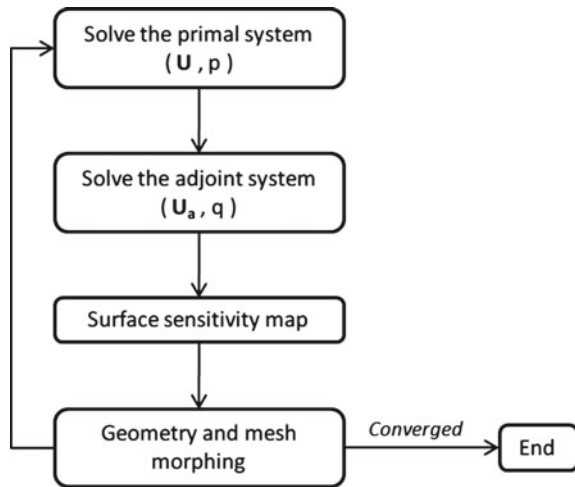


Fig. 3 Optimization loop



assessment is required for the adjoint variables. By solving both systems of equations, a surface sensitivity map can be extrapolated. The information contained in it suggests how to modify the body shape in order to increase the performance. Based on this information the geometry boundary can be modified. Eventually, as proposed in the present work (Section “[Constrained Morphing Routine for an Optimal Design](#)”), a mesh morphing solver can be included in the optimization routine to allow automatic successive steps of the optimization process. A schematic loop of the optimization routine implemented is shown in Fig. 3.

Adjoint Equations

The problem to be considered is: minimize J satisfying the Navier-Stokes equations. The variable J represents a general cost function, dependent on the flow variables \mathbf{U} and p , respectively velocity and pressure, and on the design variables.

Following the derivation in Papoutsis-Kiachagias and Giannakoglou (2016), the adjoint equations, continuity and momentum, are:

$$\nabla \cdot \mathbf{U}_a = \frac{\partial J_\Omega}{\partial p} \quad (2)$$

$$\mathbf{U}_a \cdot \nabla \mathbf{U} - \mathbf{U} \cdot (\nabla \cdot \mathbf{U}_a) = \nabla \cdot (2\nu_{eff} D(\mathbf{U}_a)) - \nabla q - \frac{\partial J_\Omega}{\partial \mathbf{U}} \quad (3)$$

The boundary conditions, derived for the following specified boundaries, are listed in Table 1.

where: \mathbf{U} and p represent the velocity and the pressure (primal variables); \mathbf{U}_a and q are the adjoint velocity and the adjoint pressure (adjoint variables); ν_{eff} is the effective viscosity; $D(\mathbf{U}_a)$ represents the rate of strain tensor; the subscripts Ω and Γ are referred, respectively, to the volume and boundary contributions of the cost function; while the subscripts t and n are referred to the tangential and normal component.

As highlighted by the adjoint equations and their boundary conditions, the adjoint problem needs to be fed with information related to the cost function, making a one-to-one relation between them. The cost function chosen for the U-bend optimization, i.e. the total pressure loss between inlet and outlet (Eq. 1), gives a contribution only to the boundary conditions, likewise all the cost functions that are only defined along the boundaries. Those cost functions identify a class for which the adjoint equations do not vary with a cost function variation.

In the present work the variation of the effective turbulent viscosity with a design modification has been considered negligible. The so-called frozen turbulence assumption neglects the change of turbulent quantities under geometry variations. The use of the frozen turbulence assumption can have a significant influence on the obtained solution (Zymaris et al. 2009). On the other hand, the use of an adjoint turbulence model would increase the complexity of the method. Despite the possible loss of accuracy, the use of this hypothesis leads to an overall validity of the solution obtained.

Table 1 Adjoint boundary conditions

Inlet and walls		
$\mathbf{U}_{ar} = 0$	$U_{an} = -\frac{\partial J_\Gamma}{\partial p}$	$n \cdot \nabla q = 0$
Outlet		
$q = U_n U_{an} + \nu_{eff} (\mathbf{n} \cdot (\nabla \cdot U_{an})) + \frac{\partial J_\Gamma}{\partial U_n}$		
$U_n \mathbf{U}_{ar} + \nu_{eff} (\mathbf{n} \cdot (\nabla \cdot \mathbf{U}_{ar})) + \frac{\partial J_\Gamma}{\partial \mathbf{U}_t} = 0$		

Surface Sensitivity Map

The adjoint optimization method uses the information from the derivative of the objective function with respect to the design variables in its search for the minimum. The design variables chosen in the present work are represented by the normal displacement of each surface node, β , resulting in a very rich design space.

The gradient information, also called surface sensitivity map, can be evaluated from the solution of the primal and adjoint equations system, as in Eq. 4.

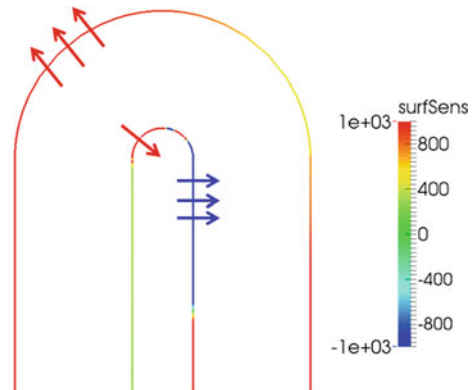
$$S = \frac{\partial J}{\partial \beta} = -v_{eff} \frac{\partial \mathbf{U}_a}{\partial \mathbf{n}} \frac{\partial \mathbf{U}}{\partial \mathbf{n}} \quad (4)$$

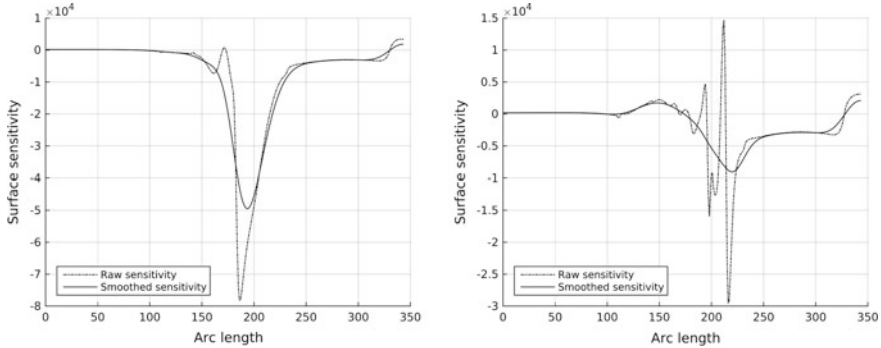
The surface sensitivity map is evaluated at the walls and expresses how much the geometry is sensitive to the performance, i.e. how much the objective function would change for a unit movement in the direction of the surface normal. A highly sensitive area entails that only a very small shape modification would already result in a large change of the objective function. The space of design variables is composed of the normal nodes displacement, thus the modifications must be performed in the direction normal to the surface itself. The sensitivity map obtained in the 2D U-bend test case is shown in Fig. 4.

A positive sensitivity region has to be moved away from the fluid to increase the performance, while a negative sensitivity value indicates that a performance improvement would be reached by moving those regions towards the fluid. On the other hand, modifications of a zero sensitivity map region will not influence the cost function.

Before converting the gradient information into a design variation, a preliminary step is needed. The surface sensitivity presents high frequency oscillations that, due to the use of the node based approach, would result in an irregular shape if not treated in a suitable way. Moreover, the high frequency oscillations will amplify during the optimization phases, causing mesh distortion problems and ultimately leading to

Fig. 4 Surface sensitivity map (2D test case)





(a) At the first loop iteration.

(b) After 100 loop iterations.

Fig. 5 Raw and smoothed sensitivity signal along the 2D U-bend inner wall

unrealistic shapes. The attainment of a smooth sensitivity is a crucial point to obtain a valid geometry deformation, and thus an admissible design. Hence, the use of a filter that takes out the high order oscillations is indispensable. The choice of the smoothing strategy requires particular attention: while it is mandatory to get rid of the discontinuities, the main information content of the sensitivity must be preserved.

A weighted average smoothing has been used in the present work for the 2D optimization test case. It corresponds to successive weighted interpolations of the surface sensitivity from face centers to boundary points and vice versa, resulting in a weighted average that takes into account a stencil of three consecutive nodes. In particular, the smoothed sensitivity \bar{S} has been evaluated as follow:

$$\bar{S}_i = \sum_j \omega_j \sum_i \omega_i S_i \quad (5)$$

where the weight ω is calculated as:

$$\omega_i = l_i / l_{sum} \quad (6)$$

In Eq. 6, l_i is the inverse distance between face centers and boundary points and l_{sum} is the sum of the inverse distances. The same definition is used for ω_j , where the distance to be considered is the one from the boundary points to the face centers.

The importance of the application of a smoothing function is highlighted in Fig. 5a, b, which show the raw and smoothed sensitivity signal along the 2D U-bend inner wall respectively after 1 and 100 iterations. The high frequency oscillations are already present at the first iteration step and become more pronounced with the iterations. Note how the smoothed signal still maintains the main information of the raw sensitivity.

A Gaussian smoothing Taubin (1995) has been used in the 3D test case, thus the smoothed sensitivity is given by:

$$\bar{S} = (I - \lambda K)S \quad (7)$$

where: I is the identity matrix; λ is a scale factor and it is a bounded positive number $0 < \lambda < 1$; $K = I - \Omega$; Ω represents the weights matrix, whose non-zero elements correspond to the design variables neighbours and are equal to the inverse distance between each node and its neighbour.

Although the performance of the two different smoothing strategies are comparable, the Gaussian smoothing resulted to be more efficient when dealing with high number of design variables and therefore used for the 3D optimization.

Constrained Morphing Routine for an Optimal Design

The last step of the implemented optimization process includes the modification of the design and a link to a mesh morpher solver to allow successive automatic steps towards the design improvement. This leads to two important issues: (1) the design modifications need to take into account particular limitations intrinsic to the chosen design, e.g. aerodynamic and geometrical constraints, and (2) the surface deformations must lead to a variation of the internal grid of the fluid domain which ensures a sufficient mesh quality to solve the RANS equations further.

Design Variation

Once the sensitivity oscillations have been ironed out, as shown in Section “[Surface Sensitivity Map](#)”, it is possible to link their information to a design variation. In the present work the design is updated using a steepest descent algorithm, as in Eq. 8:

$$\mathbf{x}_{i+i} = \mathbf{x}_i + \alpha \bar{S} \mathbf{n} \quad (8)$$

where \mathbf{x}_i and \mathbf{x}_{i+i} represent the position vector of each node before and after the boundary movement respectively. Each node is displaced in the direction normal to the surface, \mathbf{n} , with a magnitude depending on the smoothed surface sensitivity, \bar{S} . An additional parameter α is included and is defined as follows:

$$\alpha = \frac{\varepsilon}{\max \|\bar{S}_1\|} \quad (9)$$

The role of the parameter is to normalize the sensitivity, through its maximum value at the first iteration of the optimization loop, and to fix the maximum step size

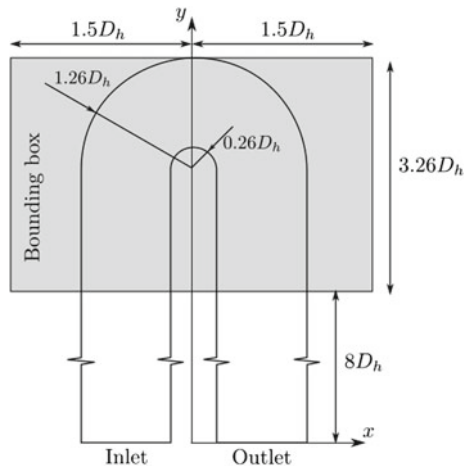
of the movement, through the parameter ε . In the present work, ε is kept constant, so it has been used a steepest descent algorithm with fixed step size. Nevertheless, the surface sensitivity decreases its magnitude as it converges to the optimum, hence the magnitude of the nodes displacement decreases iteration by iteration and eventually becomes null if a local minimum is reached.

The choice of the right value for the parameter α represents a key point of the optimization process. A big value of α could be unsuitable as it could overshoot the optimum and worsen the design straight away. On the other hand a value that is too small would exponentially increase the computational time. The choice of the suitable value to be assigned is left to the user, as it is highly case-dependent. The results shown in the present work refer to a value of $\varepsilon = 10^{-4}$ and of $\varepsilon = 5 \times 10^{-3}$, respectively, for the 2D and 3D optimization case. A bigger initial deformation has been imposed to the 3D case in order to minimize the number of optimization cycles needed to reach an optimum design, being each 3D flow field evaluation much more computationally expensive than the 2D one.

The algorithm described by Eq. 8 represents a classical steepest descent algorithm and allows a huge degree of freedom in the attainment of the optimized design. However, the majority of the designs must respect constraints, geometrical and aerodynamic, and the optimization is usually required in a limited region of the design. The planar constraints and limitations present in the U-bend test case are shown in Fig. 6. The height of the channel is allowed to change up to $0.1 D_h$.

The bounding box shown in Fig. 6 indicates the constraints to be satisfied and the zones of the U-bend that can take part in the optimization process. In order to fulfil the requirements, a constrained steepest descent algorithm has been implemented. This has been included in a versatile optimization routine that allows to the user the freedom to choose between unconstrained and constrained optimization. In the latter, the definition of a bounding box is requested. Finally, it is given to the user the possibility to choose whether to perform the optimization in a confined region or in the whole design.

Fig. 6 Planar constraints and limitations in the U-bend test case



Mesh Morphing Strategy

The last step of the optimization process is represented by a link between the boundary movement and the mesh deformation. For the cases where the motion is solution-dependent, it can be used a mesh morpher based on the Laplace equation in which the prescribed boundary movement represents the boundary condition for the internal cell motion. Details about the mesh morphing strategy can be found in Jasak and Tukovic (2006). In the present work, a distance-based quadratic method has been chosen, so the diffusivity in the Laplace equation is a function of the inverse square distance from the nearest boundary. A distance-based diffusivity improves the mesh quality near the boundaries: it redistributes the movement inside the domain allowing bigger deformations on the center of the domain where bigger cells are present.

Nevertheless, the described strategy is not sufficient to avoid a local worsening of the mesh quality near the moving boundaries. The solver calculates the cell displacement based on a cell-centred approach and interpolates to obtain the points movement. However, boundaries with prescribed movement maintain their imposed value, leading to possible differences between those values and the ones of the internal points and eventually to highly distorted cells, especially for very thin layers. In order to maintain a good mesh quality, a point interpolation method has been used (Mesh 2018). The method allows to maintain a fixed first layer height by extending the interpolation to the boundaries. The difference between the interpolated value on the boundary and the desired boundary condition is then propagated into the mesh.

The main concern about the use of a mesh movement solver is its capability to preserve the mesh quality. It is indeed of crucial importance for an automatic process to avoid the accumulation of error due to low mesh quality. The strategy described, together with the use of a suitable smoother and an appropriate step size for the boundary movement, allows the fulfilment of the mesh quality criteria. The cancellation of the high frequency noise in the sensitivity map is indeed fundamental in order to avoid overlapping of boundaries. In order to ensure the correctness of the process, thus avoiding to perform the optimization with a highly distorted mesh, a mesh quality check has been introduced after each mesh morphing step. The additional step assures that the mesh quality criteria are satisfied at each iteration step and decides whether to continue with successive iterations of the optimization loop or to end the process.

Optimization Results

The application of the steps described outlines an optimization routine that should bring to an improvement of the design with respect to the objective function taken into account. In the present work, the optimization routine has been fed with a U-bend shape with the aim to reduce the total pressure loss in it. In order to verify the reliability of the procedure developed, a 2D optimization corresponding to the

symmetry plane of the U-bend design has been performed. The satisfactory results obtained with it, allowed to proceed with the optimization of the 3D design.

2D Optimization Test Case

The outcome of the optimization routine is illustrated in Fig. 7, which shows different optimized shapes, their normalized velocity field and their velocity streamlines. Figure 7a shows a rough transition from the design regions subjected to boundaries movement and the regions that are not allowed to take part in the optimization process. Even though an improvement in the pressure drop has been achieved ($\Delta p = -20\%$), the shape obtained cannot be considered during an industrial process because of the highlighted discontinuity. The abrupt variation in the shape takes place over one single cell, hence the obtained result is questionable.

In order to overcome this problem, two approaches are proposed: the assignment of a continuity constraint for the surface sensitivity or for the boundary displacement, respectively in Fig. 7b, c. The constraints allow to obtain smooth optimized designs that could be manufactured in an industrial process. The continuity constraint imposed on the surface sensitivity is less demanding, so bigger design modifications can be reached. Despite the biggest improvement is achieved imposing the constraint on the sensitivity ($\Delta p = -31.5\%$), the constraint on the displacement seems to give results more suitable to an industrial process. Indeed fewer modifications should be performed on the original geometry, maintaining a remarkable improvement in the design performance ($\Delta p = -28.5\%$). Obviously, the attainment of the continuity constraints allows to reach better improvements, being the discontinuities present in the initial optimization (Fig. 7a) a limitation to the capability of the optimizer that remains trapped in a local minimum.

A comparison between the flow field in the original and optimized geometries, Figs. 2 and 7 respectively, highlights the direction taken by the optimization routine to achieve its objective. The body shape is modified in order to obtain a decrease of the pressure drop through the reduction of the separation bubble size that takes form after the bend. The separation region is indeed the main source of pressure loss in the proposed design, thus its suppression represents the target of the optimization routine. Analyses of the optimized U-bend shapes verify the characteristics discussed in the previous sections. A flat region is present at the top part of the bend (Fig. 7a, b), showing that the geometrical constraints imposed have been reached and satisfied. The external geometries are smooth, proving the effectiveness of the smoothing strategy used on the sensitivity. The mesh morpher has been shown to be robust enough so that the mesh quality criteria were satisfied at each iteration.

The variation of the cost function during the optimization process is shown in Fig. 8. The three different optimizations performed reach a stable minimum, identified from the attainment of a constant value of the cost function. The introduction of the continuity constraints clearly influences the optimization history, resulting in a speed up of the optimization process. The preference for the application of the

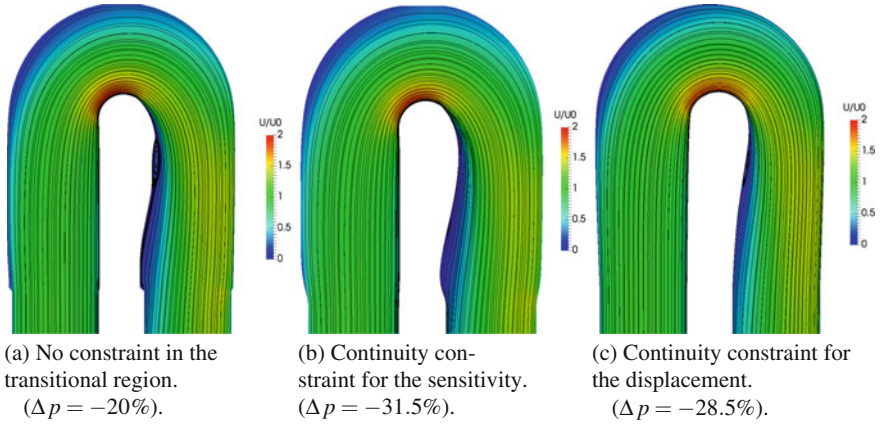


Fig. 7 Optimized 2D U-bend shape: velocity field and streamlines

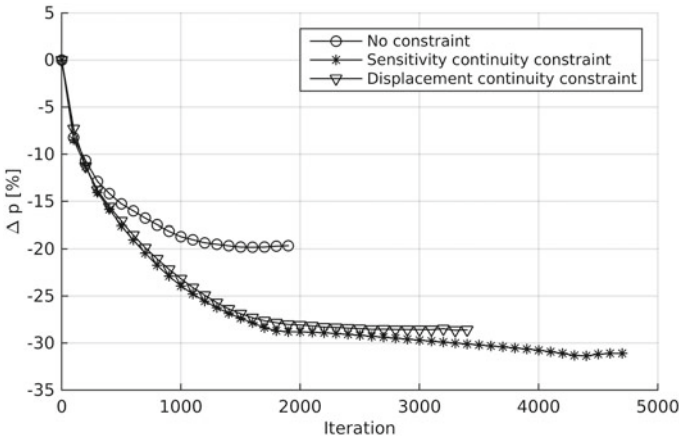


Fig. 8 Optimization history 2D U-bend

constraint on the displacement rather than on the sensitivity is confirmed also by the optimization history, since a comparable final improvement is obtained halving the computational time of the process. The optimization history of the case with a continuity constraint on the displacement converges in approximately 2000 iterations. The high number of iterations performed does not imply the use of a computational expensive routine. Indeed, the use of partially converged variables at each iteration step considerably speeds up the process, allowing to reach the minimum in a night-time period. A fully converged simulation was carried out at the end of the optimization process, confirming the 28.5% improvement in the total pressure loss and thus verifying the validity of the approach used.

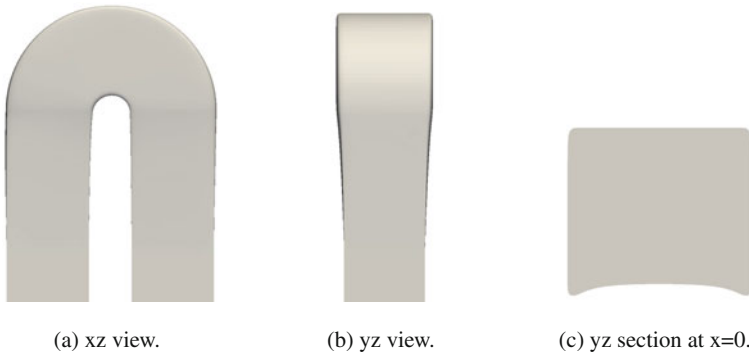


Fig. 9 Optimized 3D U-bend shape

3D Optimization Test Case

The optimization performed on the 2D test case highlighted the necessity to impose a continuity constraint in the transitional region in order to avoid abrupt variations in the design. The best result in terms of design and computational time has been attained with the assignment of the continuity constraint on the boundary displacement, thus it has been applied also in the 3D optimization case. The optimized U-bend design obtained is shown in Fig. 9, in particular a xz and yz view, (a) and (b), and a section at $x = 0$ (c). The section gradually enlarge in the y-direction until the constraint (Fig. 9b). Different xz profiles are present along the y axis as illustrated in the yz section (Fig. 9c). A smooth external shape is attained, proving once again the effectiveness of the smoothing strategy proposed. The final design obtained would be difficult to predict without the use of an optimization routine. A comparison between the resulting velocity fields, Fig. 10, and the initial ones, Fig. 1, confirms the tendency of the optimizer to modify the design with the aim to suppress the separation bubble, as observed in the 2D optimization case. Moreover, a significant reduction of the maximum velocity inside the geometry is obtained.

The route towards the optimum design is shown in Fig. 11. A stable local minimum has been reached with an improvement on the total pressure loss of the 30%. The initial bigger deformation applied to the 3D case in comparison to the 2D one allows to reach the optimum design in approximately 25 iterations and in a night-time period using 12 processors. The mesh morpher routine has been shown to be able to tolerate the big deformations imposed.

Conclusions

The optimization of an internal flow application has been described in the present work, detailing the structure of the routine developed. The U-bend test case has been optimized with respect to the total pressure loss. The use of an adjoint optimiza-

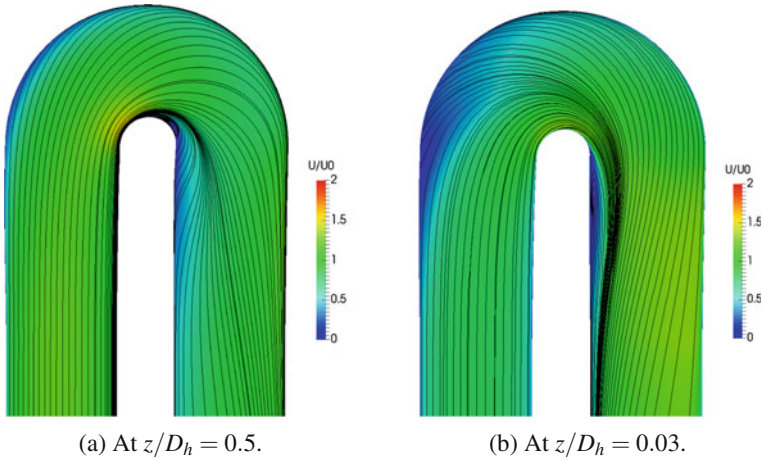


Fig. 10 Optimized 3D U-bend: velocity field and streamlines

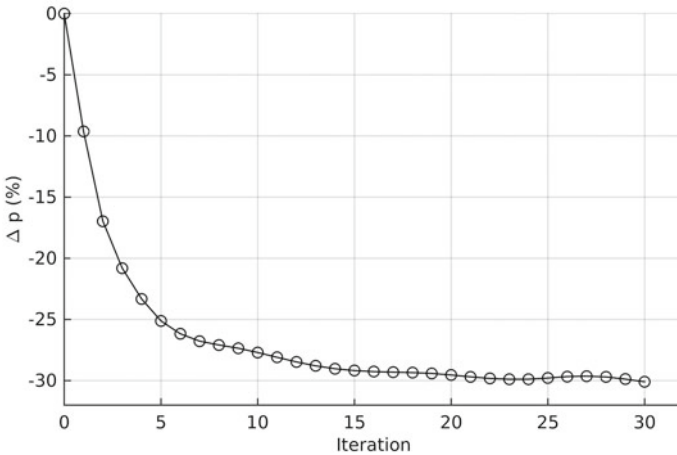


Fig. 11 Optimization history 3D U-bend

tion strategy allowed to efficiently approach the problem, given the high number of design variables taken into account. The solution of the primal and adjoint problem gives information on the modifications to perform in order to improve the design, which has been attained through the link to a constrained mesh morphing routine. The optimization has been performed in a restricted area of the design and subjected to geometrical constraints. Limitations on the regions that can be optimized are generally present, however the evaluation of an unfeasible optimization covering the whole geometry could be of interest. Indeed, it would enlarge the design space and it could lead to ideas for further designs and for different design strategies.

The optimization routine allowed to obtain an improvement of the cost function of the 28.5–30%, respectively in the 2D-3D case, through the reduction of the separation bubble size that develops after the bend. Manufacturability of the design is assured by a smooth external geometry.

Acknowledgements This work was supported by the Fonds Wetenschappelijk Onderzoek Vlaanderen—FWO. The authors want to thank the OpenCFD Limited team for the advice in the codes development.

References

- Coletti F, Verstraete T, Vanderwielen T, Bulle J, Arts T (2011) Optimization of a u-bend for minimal pressure loss in internal cooling channels—part II: experimental validation. ASME paper GT2011-46555
- Goit J, Meyers J (2014) Optimal control of wind farm power extraction in large eddy simulations. In: AIAA SciTech 32nd ASME wind energy symposium, AIAA 2014-0709
- Han J, Dutta S, Ekkad S (2000) Gas turbine heat transfer and cooling technology. Taylor and Francis, New York
- Jameson A (1988) Aerodynamic design via control theory. *J. Sci. Comput.* 3(3):233–260
- Jsak H, Tukovic Z (2006) Automatic mesh motion for the unstructured finite volume method. *Trans. FAMENA* 30(2):1–20
- Mesh motion: new point interpolation methods. <https://www.openfoam.com/releases/openfoam-v3.0+/numerics.php>
- Namgoong H, Son C, Ireland P (2008) U-bend shaped turbine blade cooling passage optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, AIAA 2008-5926
- Othmer C (2014) Adjoint methods for car aerodynamics. *J Math Ind* 4(1):6
- Papoutsis-Kiachagias E, Giannakoglou K (2016) Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications. *Arch Comput Methods Eng* 23(2):225–299
- Pironneau O (1974) On optimum design in fluid mechanics. *J Fluid Mech* 64(part I):97–110
- Taubin G (1995) Curve and surface smoothing without shrinkage. In: Proceedings of the fifth international conference on computer vision, pp 852–857
- Verstraete T, Coletti F, Bulle J, Vanderwielen T, Arts T (2011) Optimization of a u-bend for minimal pressure loss in internal cooling channels—part I: numerical method. ASME paper GT2011-46541
- Verstraete T, Li J (2013) Multi-objective optimization of a u-bend for minimal pressure loss and maximal heat transfer performance in internal cooling channels. In: ASME Turbo Expo 2013: turbine technical conference and expositions, 3A: Heat Transfer
- Zehner S, Steinbrck H, Neumann S, Weigand B (2009) The ice formation method: a natural approach to optimize turbomachinery components. *Int J Des Nat* 3(4):259–272
- Zymaris A, Papadimitriou D, Giannakoglou K, Othmer C (2009) Continuous adjoint approach to the spalart-allmaras turbulence model for incompressible flows. *Comput Fluids* 38:1528–1538

On the Properties of Solutions of the 2D Adjoint Euler Equations



Carlos Lozano

Abstract We discuss the structure of the solutions to the 2D inviscid adjoint equations on airfoils, including the behavior across shocks and sonic lines, the singularities at the forward stagnation streamline and at the trailing edges and the structure on the supersonic bubble.

Introduction

Adjoint methods are being routinely applied in optimum aerodynamic design, flow control, as well as error estimation and mesh adaptation. The solution of the adjoint equations links the sensitivity of a given cost function to perturbations of the flow. In design applications, these perturbations are shape deformations, and the adjoint solution yields the gradients of the cost function with respect to the design variables that can be fed to a gradient-based optimization algorithm. In error control applications, the adjoint method provides an estimation of the error incurred in the cost function caused by local errors in the discrete flow solution, which can be used as an indicator in a mesh adaptation algorithm.

Since the pioneering work of Jameson on aerodynamic design optimization (Jameson 1988, 1995), the focus has been set on the development of adjoint-based optimization methods, while very little attention has been paid to the structure of the adjoint solutions. Moreover, in many circumstances the adjoint solutions cannot be validated directly, and indirect validations through the sensitivities are commonplace. The problem with this approach is that an accurate adjoint solution does not guarantee accurate sensitivities and, conversely, rather accurate sensitivities can be derived from manifestly inaccurate adjoint solutions (Lozano and Ponsin 2012).

Some work has been devoted in the past to the investigation of the properties of adjoint solutions for the quasi-1D Euler equations as well as the 2D Euler equations (Giles and Pierce 1997, 1998). For the former, Giles and Pierce were able to derive

C. Lozano (✉)

Fluid Dynamics Branch, National Institute for Aerospace Technology (INTA), Madrid, Spain
e-mail: lozanorc@inta.es

closed-form analytic adjoint solutions for a cost function consisting in the integrated pressure along the duct using a Green's function approach (Giles and Pierce 2001).

In this way, it is possible to prove that for the chosen cost function, and for transonic solutions with a shock and a choked throat, the adjoint variables are continuous with zero gradient at the shock, where an adjoint shock boundary condition is required, and that there is a logarithmic singularity at the choked throat.

For 2D flows some insight was gained in Giles and Pierce (1997) using the same Green function approach, but progress was limited by the lack of closed-form analytic flow solutions for typical 2D cases. Anyhow, it was shown that for typical cost functions (lift/drag) there appears to be an inverse square-root singularity along the incoming stagnation streamline, which is clearly reflected in both numerical adjoint solutions and adjoint-adapted meshes.

Likewise, based on numerical evidence, it was claimed that there is no singularity at sonic lines (provided that they are not orthogonal to the flow, as is usually the case), and that the adjoint variables are again continuous at shocks, along which an internal adjoint boundary condition is required.

Aside from these early works, the literature is scarce in this type of theoretical analysis. Instead, adjoint solutions are usually discussed, if at all, rather descriptively, based on numerical solutions. In this work we attempt to reconcile both approaches. We will start by presenting the result of a typical numerical (inviscid) 2D adjoint solution, pointing out its most salient features, and we will try to fill in the gaps with theoretical justification where possible and applicable.

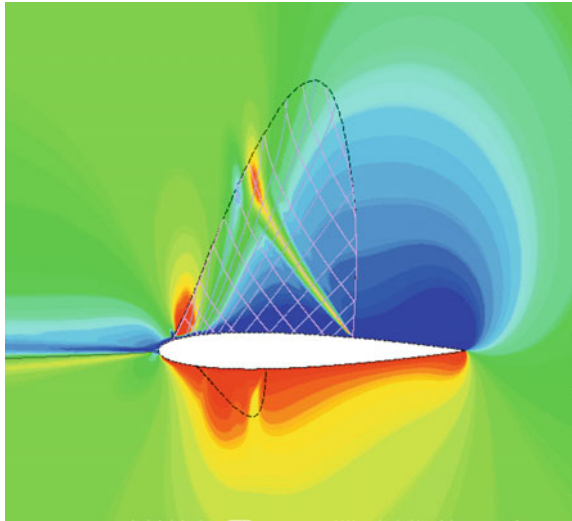
Finally, even though the properties of adjoint solutions (especially inviscid ones) are themselves of little practical relevance for design, which focuses on viscous applications and pays little attention to the details of adjoint solutions, it is still important, from a fundamental viewpoint, to characterize precisely the behavior of inviscid adjoint solutions for both validation of numerical solvers and a deeper understanding of the adjoint equations. Likewise, the structure of the adjoint solution is relevant to adjoint-based mesh adaptation algorithms, as new nodes in the adapted meshes tend to cluster at regions with large adjoint gradients or adjoint singularities.

A Typical (Inviscid) 2D Adjoint Solution

Figure 1 shows a typical (drag) adjoint solution for transonic flow past a NACA0012 airfoil with $M=0.8$ and $\alpha = 1.25^\circ$. This is a shocked case which will allow us to probe the most salient features of the adjoint solution. The numerical solution has been obtained with DLR's unstructured solver Tau (Schwamborn et al. 2006).

We show here a contour plot of the density adjoint. The stagnation streamline, the sonic lines delimiting the supersonic region as well as the characteristic lines (see Anderson 1990, Chap. 11) in the supersonic bubble of the baseline flow are also shown. This is obviously not the analytic adjoint solution, but continuous and discrete adjoint solutions show a very similar behavior, so the structures that we see

Fig. 1 Contour plot of the density adjoint for transonic flow past a NACA0012 airfoil with $M_\infty = 0.8$ and $\alpha = 1.25^\circ$



are likely to be, in general, reasonable approximations to those in the exact solution. There are various salient features worth-mentioning.

- The adjoint solution appears to be smooth at the shock and sonic lines, although something is clearly going on around the region where the sonic lines impact the airfoil surface (see also Fig. 2).
- The stagnation streamline is clearly visible.
- In the supersonic bubble, the characteristic line that impacts on the shock foot (corresponding to the supersonic region of the flow with the highest influence on the shock foot) can be clearly spotted in the adjoint map (Sartor et al. 2015).

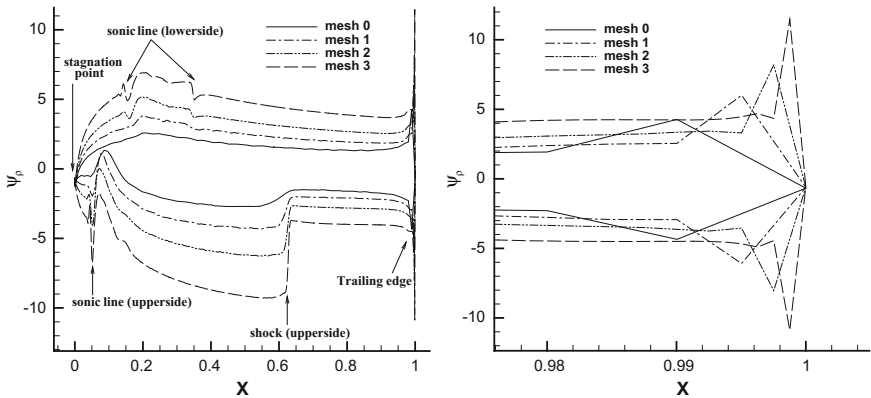


Fig. 2 Left: Adjoint density solution on the airfoil surface on four meshes. Right: zoom near the trailing edge

Additionally, Fig. 2 shows a plot of the adjoint solution on the airfoil surface on 4 sequentially finer meshes. There are three salient features in that figure:

- There is a very strong mesh dependence of the adjoint surface values, with some regions even showing hints of lack of mesh convergence or even consistency errors in the limit of increasing mesh resolution.
- This is particularly noticeable around the sonic points and at the shock location, where mesh refinement reveals the formation of a shock-like layer. This structure is likely to be related to the intersection of the supersonic characteristic line impacting the shock foot rather than to the shock discontinuity itself. This idea can be confirmed in Fig. 10, which shows that, for the same transonic NACA0012 case with $\alpha = 0^\circ$, both the characteristic line and the layer are missing from the adjoint field. Notice that, at any rate, the above behaviour is, however, strongly case dependent. Figure 10 itself shows proof of that, with the mesh dependence restricted to the vicinity of the sonic points.
- Finally, there is a singularity at the trailing edge, where the values of the adjoint variables at the next-to-trailing-edge nodes grow continually as the mesh is refined. We will come back to this issue in Section “Trailing Edge Singularity”.

Behavior at Shocks

The analysis of the 2D Adjoint Euler equations with shocks has been carried out in quite some detail in Baeza et al. (2009). Here, we will just review the chief results and build on them in order to derive a few results.

Suppose then that we have transonic inviscid flow past an airfoil profile S on a domain Ω with far-field boundary S^∞ . Suppose also that there is an attached shock with profile Σ extending from x_b (shock foot) to x_{end} (shock tip) (Fig. 3).

Let us assume that we want to compute the sensitivity derivatives of the following cost function

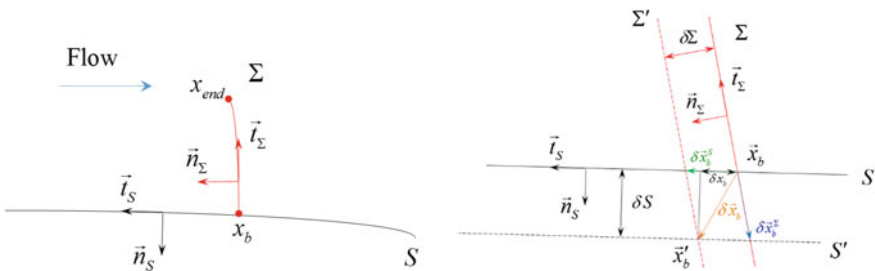


Fig. 3 Scheme of shock location and conventions

$$J(S) = \int_{S \setminus x_b} h(p, \vec{n}_S) dS \quad (1)$$

with respect to variations of the shape of S . (In (1), p is the fluid pressure, in such a way that when $h(p, \vec{n}_S) = p(\vec{n}_S \cdot \vec{d})$, J corresponds to the force exerted by the fluid on the surface S measured along a direction \vec{d}). A small deformation of S , $\vec{x}_S \rightarrow \vec{x}_S + \delta\vec{x}_S$ causes a flow perturbation δU , where $U = (\rho, \rho\vec{v}, \rho E)^T$ (with ρ, \vec{v}, E the fluid's density, velocity and total energy, respectively) is the vector of conservative variables. In the perturbed flow the shock structure and position will be different, and we will assume that the new shock curve can be described in terms of a local (small) deformation $\vec{x}_\Sigma \rightarrow \vec{x}_\Sigma + \delta\vec{x}_\Sigma$. As a result of these perturbations the cost function changes too, and the corresponding linearized perturbation can be computed with the adjoint method. Introducing adjoint states $\psi = (\psi_1, \psi_2, \psi_3, \psi_4)^T$ and $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)^T$ to enforce the (Euler) flow equations $\nabla \cdot \vec{F} = 0$ and the (Rankine-Hugoniot) shock equations $[\vec{F} \cdot \vec{n}_\Sigma]_\Sigma = 0$ (where $\vec{F} = (\rho\vec{v}, \rho\vec{v}\vec{v}_x + p\hat{x}, \rho\vec{v}\vec{v}_y + p\hat{y}, \rho\vec{v}H)^T$ is the flux vector and H the total enthalpy) and $[(\cdot)]_\Sigma = (\cdot)_{downstr} - (\cdot)_{upstr}$ denotes the jump across the shock), linearizing the resulting cost function and rearranging yields

$$\begin{aligned} \delta J(S) &= \int_{S \setminus x_b} \partial_p h(p, \vec{n}_S) (\delta\vec{x}_S \cdot \nabla p) dS + \int_{S \setminus x_b} \nabla_{\vec{n}_S} h(p, \vec{n}_S) \cdot \delta\vec{n}_S dS \\ &+ \int_{S \setminus x_b} h(p, \vec{n}_S) (\partial_{t_g} \delta S_t - \delta S_n \kappa_S) dS + \int_{S \setminus x_b} \partial_p h(p, \vec{n}_S) \delta p dS \\ &+ \left(\frac{(\vec{n}_S \cdot \vec{n}_\Sigma) \delta S_n - \delta \Sigma_n}{(\vec{n}_S \cdot \vec{t}_\Sigma)} \right)_{x_b} [h(p, \vec{n}_S)]_{x_b} + \int_{\Omega \setminus \Sigma} \nabla \psi^T \cdot \vec{F}_U \delta U d\Omega \\ &- \int_{S \setminus x_b} \psi^T (\vec{F}_U \cdot \vec{n}_S) \delta U dS - \int_{S^\infty} \psi^T (\vec{F}_U \cdot \vec{n}_{S^\infty}) \delta U dS^\infty - \int_{\Sigma} [\psi^T \vec{F}_U \delta U]_\Sigma \cdot \vec{n}_\Sigma d\Sigma \\ &- \int_{\Sigma} \theta^T [\vec{F}_U \delta U]_\Sigma \cdot \vec{n}_\Sigma d\Sigma - \int_{\Sigma} \theta^T [\vec{F} \cdot \delta\vec{n}_\Sigma]_\Sigma d\Sigma - \int_{\Sigma} \theta^T [(\delta\vec{x}_S \cdot \nabla) \vec{F} \cdot \vec{n}_\Sigma]_\Sigma d\Sigma \quad (2) \end{aligned}$$

where $\delta S_n = \vec{n}_S \cdot \delta\vec{x}_S$ and $\delta S_t = \vec{t}_S \cdot \delta\vec{x}_S$ are the normal and tangent parts of the shape deformation and κ_S is the local curvature of the profile, while $[\cdot]_{x_b}$ is the jump across the shock at the shock foot. Also, $\delta\vec{x}_\Sigma = \delta\Sigma_t \vec{t}_\Sigma + \delta\Sigma_n \vec{n}_\Sigma$ and we have already integrated by parts the term $-\int_{\Omega \setminus \Sigma} \psi^T \nabla \cdot (\vec{F}_U \delta U) d\Omega$. We now use that $\delta\vec{n}_\Sigma = -(\kappa_\Sigma \delta\Sigma_t + \partial_{t_g} \delta\Sigma_n) \vec{t}_\Sigma$, κ_Σ is the local curvature of the shock profile, and that $\psi^T (\vec{F}_U \cdot \vec{n}_S) \delta U = (\vec{\varphi} \cdot \vec{n}_S) \delta p + \rho \delta \vec{v} \cdot \vec{n}_S (\psi_1 + \vec{\varphi} \cdot \vec{v} + \psi_4 H)$, where $\vec{\varphi} = (\psi_2, \psi_3)$, and integrate by parts $\int_{\Sigma} \theta^T [\vec{F} \cdot \vec{t}_\Sigma]_\Sigma \partial_{t_g} \delta\Sigma_n d\Sigma$ and $-\int_{\Sigma} \delta\Sigma_t \theta^T [\partial_{t_g} \vec{F} \cdot \vec{n}_\Sigma]_\Sigma d\Sigma$ along Σ , using the Rankine-Hugoniot condition $[\vec{F} \cdot \vec{n}_\Sigma]_\Sigma = 0$, the identities $[\vec{F} \cdot \vec{t}_\Sigma]_{x_{end}} = 0$ (by continuity) and $[\partial_{t_g} \vec{F} \cdot \vec{t}_\Sigma + \partial_{n_\Sigma} \vec{F} \cdot \vec{n}_\Sigma]_\Sigma = [\nabla \cdot \vec{F}]_\Sigma = 0$ (since $\nabla \cdot \vec{F} = 0$ on both sides of the shock), as well as the geometric identities $\partial_{t_g} \vec{t}_\Sigma = \kappa_\Sigma \vec{n}_\Sigma$, $\partial_{t_g} \vec{n}_\Sigma = -\kappa_\Sigma \vec{t}_\Sigma$. Finally, we use the perturbed non-transpiration boundary condition $\delta(\vec{v} \cdot \vec{n}_S) = \delta\vec{v} \cdot \vec{n}_S + (\delta\vec{x}_S \cdot \nabla) \vec{v} \cdot \vec{n}_S + \vec{v} \cdot \delta\vec{n}_S = 0$. This gives

$$\begin{aligned}
\delta J(S) = & \int_{S \setminus x_b} \partial_p h(p, \vec{n}_S) (\delta \vec{x}_S \cdot \nabla p) dS + \int_{S \setminus x_b} \nabla_{\vec{n}_S} h(p, \vec{n}_S) \cdot \delta \vec{n}_S dS \\
& + \int_{S \setminus x_b} h(p, \vec{n}_S) (\partial_{t_g} \delta S_t - \delta S_n \kappa_S) dS + \left(\frac{(\vec{n}_S \cdot \vec{t}_\Sigma)}{(\vec{n}_S \cdot \vec{t}_\Sigma)} \delta S_n \right)_{x_b} [h(p, \vec{n}_S)]_{x_b} \\
& + \int_{S \setminus x_b} \rho ((\delta \vec{x}_S \cdot \nabla) \vec{v} \cdot \vec{n}_S + \vec{v} \cdot \delta \vec{n}_S) (\psi_1 + \vec{\varphi} \cdot \vec{v} + \psi_4 H) dS + \int_{\Omega \setminus \Sigma} \nabla \psi^T \cdot \vec{F}_U \delta U d\Omega \\
& + \int_{S \setminus x_b} (\partial_p h(p, \vec{n}_S) - (\vec{\varphi} \cdot \vec{n}_S)) \delta p dS - \int_{S^\infty} \psi^T (\vec{F}_U \cdot \vec{n}_{S^\infty}) \delta U dS^\infty \\
& - \int_{\Sigma} \left((\psi^T + \theta^T) \vec{F}_U \delta U \Big|_{down} - (\psi^T + \theta^T) \vec{F}_U \delta U \Big|_{up} \right) \cdot \vec{n}_\Sigma d\Sigma \\
& - \int_{\Sigma} \partial_{t_g} \theta^T [\vec{F} \cdot \vec{t}_\Sigma]_\Sigma \delta \Sigma_n d\Sigma - \left(\frac{[h(p, \vec{n}_S)]_\Sigma + \theta^T [\vec{F} \cdot \vec{t}_\Sigma]_\Sigma}{(\vec{n}_S \cdot \vec{t}_\Sigma)} \right)_{x_b} \delta \Sigma_n(x_b) \quad (3)
\end{aligned}$$

Independence of (3) from δU and $\delta \vec{x}_\Sigma$ can be achieved if the flow obeys the Euler+Rankine-Hugoniot equations, the adjoint state obeys the adjoint equation

$$\nabla \psi^T \cdot \vec{A} = 0 \quad \text{in } \Omega \setminus \Sigma \quad (4)$$

(where $\vec{A} = \vec{F}_U$ are the flow Jacobians) with the following wall and far-field boundary conditions

$$\begin{aligned}
\vec{\varphi} \cdot \vec{n}_S &= \partial_p h(p, \vec{n}_S) \quad \text{on } S \setminus x_b \\
\psi^T (\vec{F}_U \cdot \vec{n}_{S^\infty}) \delta U &= 0 \quad \text{on } S^\infty
\end{aligned} \quad (5)$$

and is continuous across the shock

$$\psi|_{\Sigma^{up}} = \theta = \psi|_{\Sigma^{down}}, \quad (6)$$

where it must obey an internal shock equation

$$\begin{aligned}
\partial_{t_g} \psi^T [\vec{F} \cdot \vec{t}_\Sigma]_\Sigma &= \vec{t}_\Sigma \cdot \nabla \psi^T [\vec{F} \cdot \vec{t}_\Sigma]_\Sigma = \\
[\rho]_\Sigma v_t (\partial_{t_g} \psi_1 + H \partial_{t_g} \psi_4) &+ ([p]_\Sigma + [\rho]_\Sigma v_t^2) \vec{t}_\Sigma \cdot \partial_{t_g} \vec{\varphi} = 0
\end{aligned} \quad (7)$$

along the shock, and

$$\psi^T(x_b) [\vec{F} \cdot \vec{t}_\Sigma]_{x_b} = \frac{[h(p, \vec{n}_S)]_{x_b}}{(\vec{n}_S \cdot \vec{t}_\Sigma)_{x_b}} \quad (8)$$

at the shock foot x_b . In that case, the perturbed objective function can be computed from the remaining terms of (3) as

$$\begin{aligned}
 \delta J(S) = & \int_{S \setminus x_b} \partial_p h(p, \vec{n}_S) (\delta \vec{x}_S \cdot \nabla p) dS + \int_{S \setminus x_b} \nabla_{\vec{n}_S} h(p, \vec{n}_S) \cdot \delta \vec{n}_S dS \\
 & + \int_{S \setminus x_b} h(p, \vec{n}_S) (\partial_{t_g} \delta S_t - \delta S_n \kappa_S) dS \\
 & + \int_{S \setminus x_b} \rho ((\delta \vec{x}_S \cdot \nabla) \vec{v} \cdot \vec{n}_S + \vec{v} \cdot \delta \vec{n}_S) (\psi_1 + \vec{\varphi} \cdot \vec{v} + \psi_4 H) dS \\
 & + (\delta \vec{x}_S \cdot \vec{n}_S)_{x_b} \frac{(\vec{n}_S \cdot \vec{n}_\Sigma)_{x_b}}{(\vec{n}_S \cdot \vec{t}_\Sigma)_{x_b}} [h(p, \vec{n}_S)]_{x_b}
 \end{aligned} \tag{9}$$

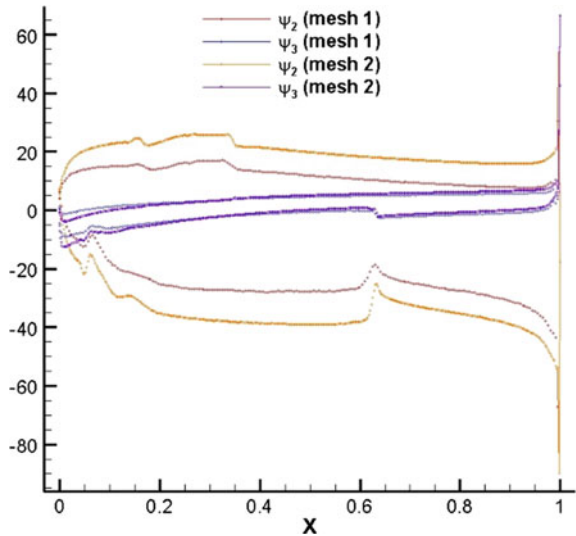
Equations (4)–(8) have the following main consequences:

- For **non-linear** cost functions, the adjoint cannot be continuous at the shock foot. This is evident already from Eq. (5), from where it follows, taking differences on both sides of the shock, that $[\vec{\varphi} \cdot \vec{n}_S]_{x_b} = [\partial_p h(p, \vec{n}_S)]_{x_b}$, so if h is a non-linear function of the pressure, then $[\partial_p h(p, \vec{n}_S)]_{x_b} \neq 0$ and thus $[\vec{\varphi} \cdot \vec{n}_S]_{x_b} \neq 0$. A numerical example of this situation can be seen in Fig. 4, where a glitch in the momentum adjoint variables $\vec{\varphi} = (\psi_2, \psi_3)$ for $J = \frac{1}{2} \int_S p^2 ds$ can be observed at the shock location.

Using the adjoint equation

$$\vec{A}^T \cdot \nabla \psi = \vec{A}^T \cdot \vec{t}_\Sigma \partial_{t_g} \psi + \vec{A}^T \cdot \vec{n}_\Sigma \partial_n \psi = 0 \tag{10}$$

Fig. 4 $\vec{\varphi} = (\psi_2, \psi_3)$ at the wall on meshes 1 and 2 for $J = \frac{1}{2} \int_S p^2 ds$



on both sides of the shock, it can be shown that the above equations entail for the normal derivatives across the shock the following conditions:

$$\begin{aligned}
 [n_\Sigma^x \partial_n \psi_2 + n_\Sigma^y \partial_n \psi_3]_\Sigma &= 0 \\
 [\partial_n \psi_1]_\Sigma + H[\partial_n \psi_4]_\Sigma + v_t [t_\Sigma^x \partial_n \psi_2 + t_\Sigma^y \partial_n \psi_3]_\Sigma &= 0 \\
 [v_n \partial_n \psi_1]_\Sigma &= 0 \\
 [v_n \partial_n \psi_4]_\Sigma &= 0
 \end{aligned} \tag{11}$$

plus various others relating the jump in adjoint normal derivatives across the shock to adjoint tangent derivatives. In (11), $v_n = \vec{v} \cdot \vec{n}_\Sigma$ and $v_t = \vec{v} \cdot \vec{t}_\Sigma$ are the normal and tangent components of the flow velocity across the shock.

- For **normal** shocks, $v_t = 0$ and is reduced to

$$t_\Sigma^x \partial_{t_g} \psi_2 + t_\Sigma^y \partial_{t_g} \psi_3 = 0 \tag{12}$$

while the wall b.c. yields now $(\vec{\varphi} \cdot \vec{n}_S)_{x_b} = [\partial_p h(p, \vec{n}_S)]_{x_b} / [p]_{x_b}$, which is only consistent with (5) if h is a linear function of the pressure. Using (12) and the adjoint equation to solve for $\partial_n \psi$ in terms of $\partial_{t_g} \psi$ yields, after some algebraic manipulations, the following result

$$\begin{aligned}
 \partial_n \psi_1 &= 0 \\
 n_\Sigma^x \partial_n \psi_2 + n_\Sigma^y \partial_n \psi_3 &= 0 \\
 \partial_n \psi_4 &= 0 \\
 t_\Sigma^x \partial_n \psi_2 + t_\Sigma^y \partial_n \psi_3 &= -\frac{1}{v_n} (\partial_{t_g} \psi_1 + H \partial_{t_g} \psi_4) \\
 &\quad - (n_\Sigma^x \partial_{t_g} \psi_2 + n_\Sigma^y \partial_{t_g} \psi_3)
 \end{aligned} \tag{13}$$

on either side of the shock. Hence, the normal gradient of the adjoint variables verifies

$$\begin{aligned}
 [\partial_n \psi_1]_\Sigma &= 0 \\
 [\partial_n \psi_2]_\Sigma &= -[v_n^{-1}]_\Sigma \partial_{t_g} (\psi_1 + H \psi_4) t_\Sigma^x \\
 [\partial_n \psi_3]_\Sigma &= -[v_n^{-1}]_\Sigma \partial_{t_g} (\psi_1 + H \psi_4) t_\Sigma^y \\
 [\partial_n \psi_4]_\Sigma &= 0
 \end{aligned} \tag{14}$$

across a normal shock.

What does one actually see in an actual numerical computation? Figures 5 and 6 show the behavior of an adjoint solution at a normal shock in a transonic case and at a bow shock in a supersonic case. Adjoint variables and their normal derivatives do appear to be continuous across **all** shocks. Additionally, adjoint normal derivatives do seem to vanish across **all** shocks, but the evidence is not conclusive. At any rate,

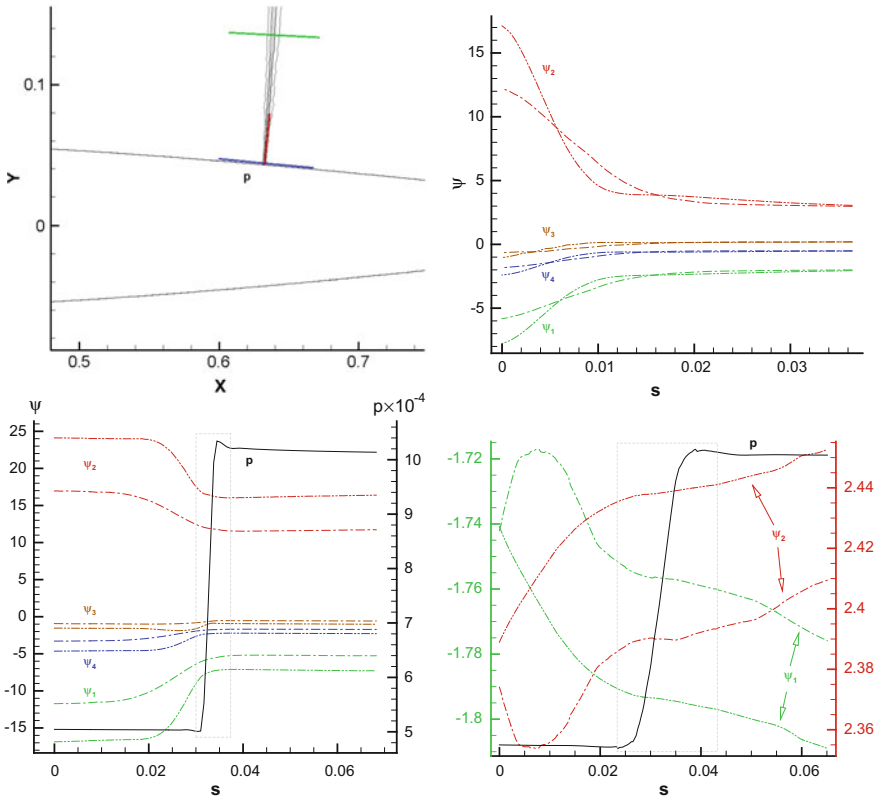


Fig. 5 Top left: Shock region and cut lines in a transonic NACA0012 case with $M_\infty = 0.8$ and $\alpha = 1.25^\circ$. Top right: Drag adjoint variables on 2 meshes along the red cut line within the shock layer and perpendicular to the airfoil. s measures distance along the line starting at the airfoil surface. Bottom left: Drag adjoint variables on 2 meshes along the blue cut line crossing the shock parallel to the airfoil surface near the shock foot. s measures distance along the line starting from the upstream side of the shock. Bottom right: Drag adjoint variables on 2 meshes along the green cut line crossing the shock at right angle. s measures distance along the line starting from the upstream side of the shock

it must be kept in mind that the shocks are under resolved and, likewise, none of the adjoint computations discussed here enforce the shock boundary conditions.

Sonic Line

Numerical computations show no sign of singular behavior at the sonic line for 2D cases. This result can be put on a more sound basis by considering the adjoint equation in coordinates orthogonal (n) and tangent (s) to the sonic line.

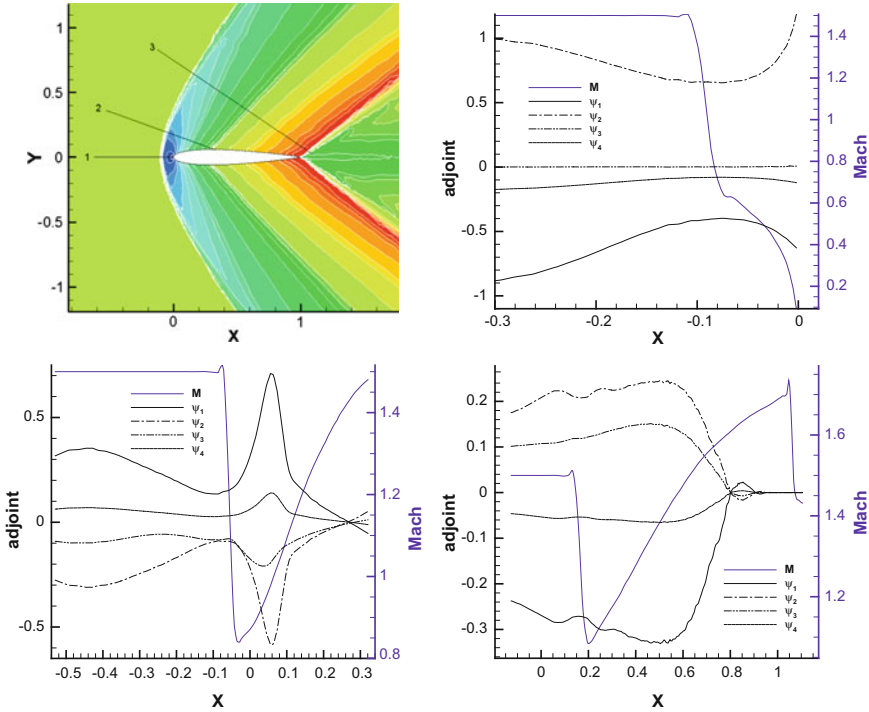


Fig. 6 Adjoint variables along a cut lines 1, 2 and 3 (top left figure) in a supersonic NACA0012 case with $M_\infty = 1.5$ and $\alpha = 0^\circ$. Line 1 (top right figure) cuts the bow shock along the centerline (thus resulting in a normal shock), while line 2 and 3 cut the bow shock at right angles away from the centerline resulting in oblique shocks

$$\vec{A}^T \cdot \nabla \psi = \vec{A}^T \cdot \vec{t} \partial_s \psi + \vec{A}^T \cdot \vec{n} \partial_n \psi = 0 \quad (15)$$

Using (15) we can solve for $\partial_n \psi$ in terms of $\partial_s \psi$. We also use that, near the sonic line, $M = 1 + M_1 n + O(n^2)$, where M_i will in general depend on s , and assume that the total enthalpy is constant ($H = H_0$) throughout. When the local flow is orthogonal to the sonic line we get from

$$\partial_n \psi = -(\vec{A}^T \cdot \vec{n})^{-1} (\vec{A}^T \cdot \vec{t}) \partial_s \psi = (n^{-1} M_{-1} + M_0 + n M_1 + \dots) \partial_s \psi \quad (16)$$

where the coefficient matrices M_i depend on the transverse coordinate s . Notice that in this case the normal Jacobian $\vec{A}^T \cdot \vec{n}$ is singular at the sonic line (the eigenvalue $\vec{v} \cdot \vec{n} - c$, where c is the soundspeed, vanishes), which is reflected in the n^{-1} singularity in (16). Hence, there is a potential logarithmic singularity in the adjoint variables across the sonic line. On the other hand, when the local flow is not orthogonal to the sonic line, we get

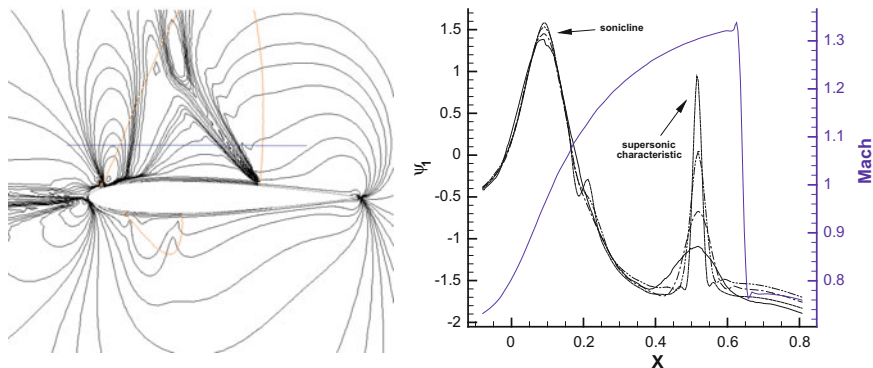


Fig. 7 Left: ψ_1 contours for transonic flow past a NACA0012 airfoil with $M_\infty = 0.8$ and $\alpha = 0$. Right: Mesh convergence plot of ψ_1 along a line crossing the supersonic bubble as indicated in the left figure

$$\partial_n \psi = -(\vec{A}^T \cdot \vec{n})^{-1} (\vec{A}^T \cdot \vec{t}) \partial_s \psi = (M_0 + nM_1 + \dots) \partial_s \psi \quad (17)$$

and the normal derivative of the adjoint across the sonic line is perfectly smooth (Fig. 7).

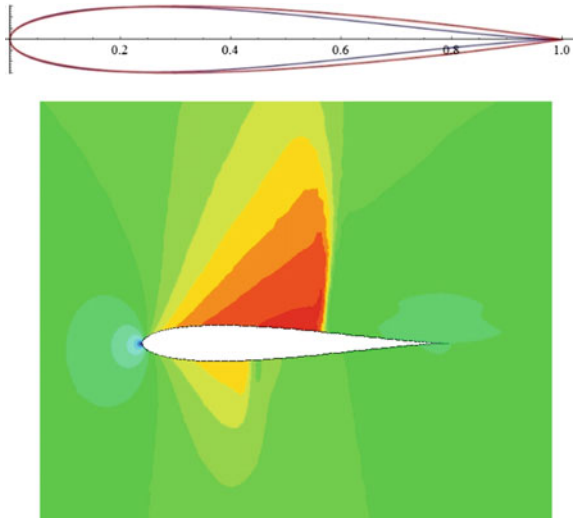
Trailing Edge Singularity

We have seen in Fig. 2 that for the transonic NACA0012 case at $M_\infty = 0.8$ and $\alpha = 1.25^\circ$, there appears to be a singularity in the adjoint variables towards the trailing edge. The tangency condition for inviscid flows on walls entails that there should be a stagnation point at finite-angle trailing edges, while at cusped trailing edges this needs not be the case. Now, Euler solvers do not obey the first condition, so one could wonder whether the observed behavior is inherent in the adjoint equations or rather it is related to the failure of the b.c. at the trailing edge. To this end, we examine the transonic flow with $M = 0.8$ and $\alpha = 1.25^\circ$ past a symmetric Joukowski airfoil resembling the NACA0012 airfoil (Fig. 8).

As can be seen in Fig. 9, the adjoint solution still has a pronounced peak towards the trailing edge that grows continually as the mesh is refined. On the other hand, the shock region is almost uneventful. Sonic lines, however, are clearly reflected.

It appears, thus, that the trailing edge singularity does not depend on the wedge angle. On the other hand, reducing the angle of attack does seem to reduce (and even eliminate) the singularity, as can be seen in Fig. 10, where the surface value of the adjoint density for a NACA0012 case with $M = 0.8$ and $\alpha = 0^\circ$ on three sequentially finer meshes is presented, clearly showing a strikingly different behavior at the trailing edge. More intuition can be gained by comparing the patterns of the adjoint momentum vector $\vec{\varphi} = (\psi_2, \psi_3)$ for both transonic NACA0012 cases and the

Fig. 8 Top: Symmetric Joukowski airfoil with 12% width (blue) against NACA0012 (red). Bottom: Mach contours for transonic flow past a symmetric Joukowski airfoil with $M_\infty = 0.8$ and $\alpha = 1.25^\circ$



symmetric Joukowski airfoil. This is done in Fig. 11, where the adjoint momentum “flow” associated with the singular behavior at the trailing edge turns around the trailing edge, unlike the flow associated with the non-singular case that shows a much more symmetric pattern.

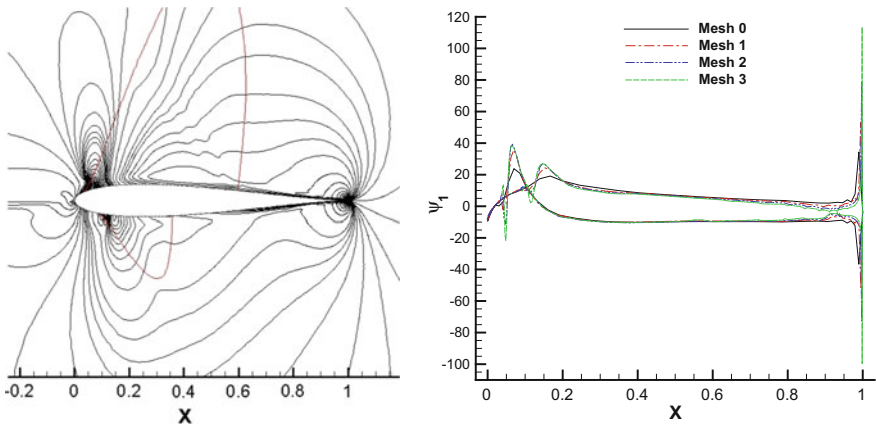


Fig. 9 Left: ψ_1 contours and sonic line (red) for transonic flow past a symmetric Joukowski airfoil with $M_\infty = 0.8$ and $\alpha = 0$. Right: ψ_1 on the airfoil surface on four sequentially finer meshes

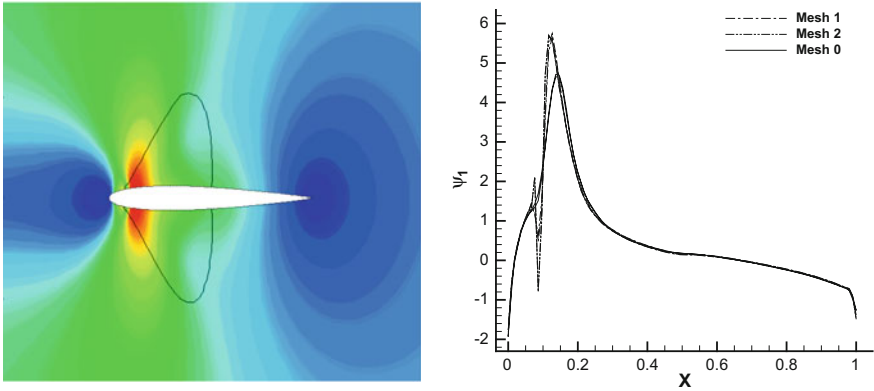


Fig. 10 Left: Contours of density adjoint variable ψ_1 for a NACA0012 airfoil with $M_\infty = 0.8$ and $\alpha = 0$. Right: Density adjoint variable ψ_1 on the airfoil surface

Supersonic Characteristic

Figure 1 shows that the supersonic characteristic impinging on the shock foot leaves a strong footprint on the adjoint solution, which actually grows continually as the mesh is refined (Fig. 7). This behavior is however not universal, as there seems to be no trace of the supersonic characteristic in either the symmetric ($\alpha = 0^\circ$) transonic NACA0012 case (Fig. 10) or in the transonic symmetric Joukowski airfoil (Fig. 9). Notice that in those cases, no layer appears to form at the shock location in the surface adjoint plots, thus suggesting that such behavior is likely tied to the adjoint structure around the supersonic characteristic.

Stagnation Streamline

The behavior at the stagnation streamline is singular as anticipated in Giles and Pierce (1997), including the expected inverse square-root character of the singularity (Fig. 12).

Conclusions

We have presented a number of ideas concerned with the structure of the solution of the adjoint 2D Euler equations. These include:

- The analysis of the consequences of the adjoint shock equations. These include continuity of adjoint variables across shocks and a number of relations for the

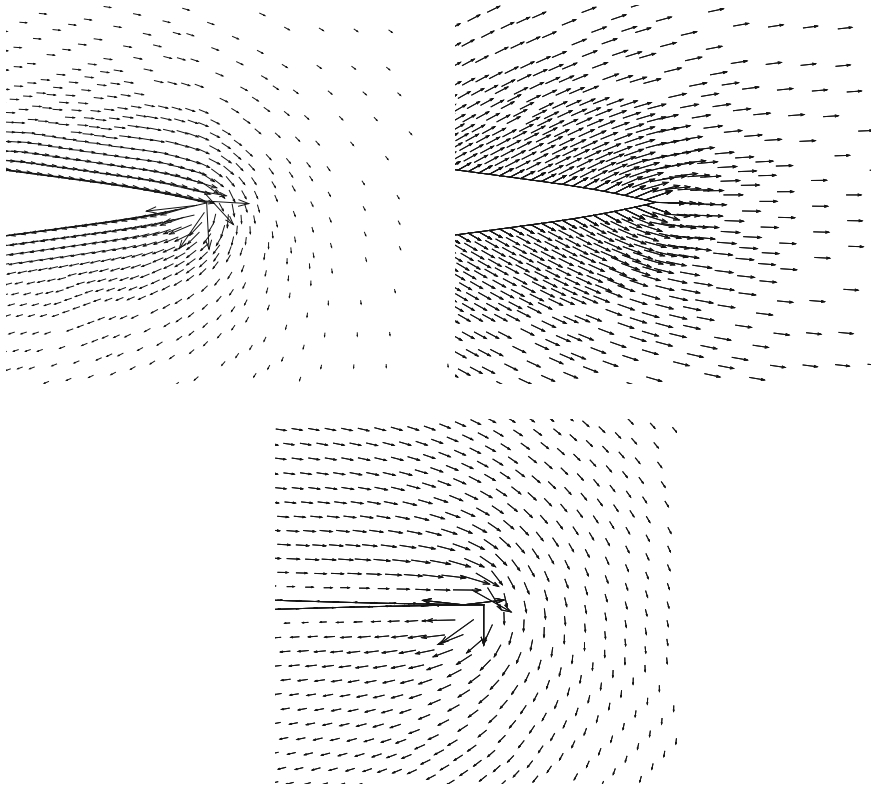


Fig. 11 Pattern of the adjoint momentum vector $\vec{\varphi} = (\psi_2, \psi_3)$ near the trailing edge for $M_\infty = 0.8$ transonic NACA0012 cases with $\alpha = 1.25^\circ$ (top left) and $\alpha = 0$ (top right) and the symmetric Joukowski airfoil with $M_\infty = 0.8$ and $\alpha = 1.25^\circ$ (bottom)

adjoint normal derivatives across the shock. For normal shocks, these relations allow to prove that normal derivatives are mostly vanishing (and continuous) across **normal** shocks, while for oblique shocks the situation is not clear. In all cases, however, numerical evidence seems to support continuous, vanishing normal derivatives across all shocks.

- The analytic confirmation that no singularity occurs across sonic lines unless the flow is locally orthogonal to the sonic line.
- A further analysis of the behavior of the (computed) inviscid adjoint solution near the trailing edge of airfoils, including the confirmation that a singularity (which hints at a consistency error in the limit of increasing grid resolution) is present even for cusped trailing edges and is related to the adjoint momentum “flow” turning around the trailing edge.

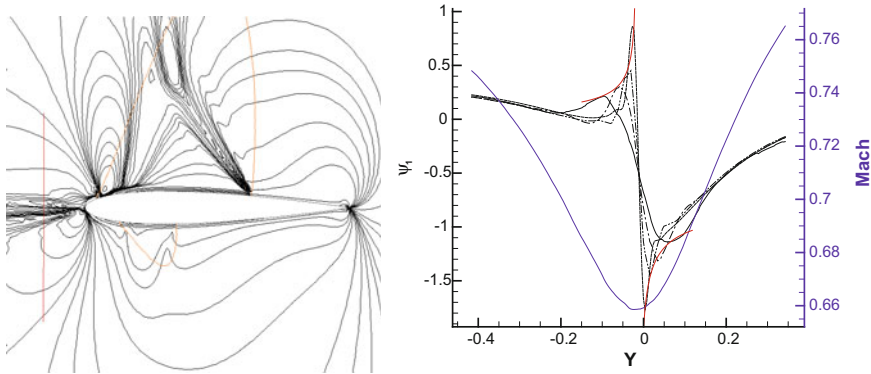


Fig. 12 Left: ψ_1 contours for transonic flow past a NACA0012 airfoil with $M_\infty = 0.8$ and $\alpha = 1.25^\circ$ Right: Mesh convergence plot of ψ_1 along a line crossing the stagnation streamline as indicated in the left figure. Fits to functions $a + b/\sqrt{|y - y_0|}$, which confirm the predicted inverse square-root behavior of the singularity, are shown in red

References

- Anderson JD (1990) Modern compressible flow, McGraw-Hill, 2nd edn
- Baeza A, Castro C, Palacios F, Zuazua E (2009) 2-D Euler shape design on nonregular flows using adjoint Rankine-Hugoniot relations. *AIAA J* 47(3):552–562
- Giles MB, Pierce NA (1997) Adjoint equations in CFD: duality, boundary conditions and solution behavior. *AIAA Paper* 97–1850
- Giles MB, Pierce NA (1998) On the properties of solutions of the adjoint Euler equations. In: Baines M (ed) Numerical methods for fluid dynamics VI, IFCD, pp 1–16
- Giles MB, Pierce NA (2001) Analytic adjoint solutions for the quasi-one-dimensional Euler equations. *J Fluid Mech* 426:327–345
- Jameson A (1988) Aerodynamic Design via Control Theory. *J Sci Comp* 3:233–260
- Jameson A (1995) Optimum Aerodynamic Design Using CFD and Control Theory, *AIAA Paper* 95–1729
- Lozano C, Ponsin J (2012) Remarks on the numerical solution of the adjoint quasi-one-dimensional Euler equations. *Int J Numer Meth Fluids* 69(5):966–982
- Sartor F, Mettot C, Sipp D (2015) Stability, receptivity, and sensitivity analyses of buffeting transonic flow over a profile. *AIAA J* 53(7):1980–1993
- Schwamborn D, Gerhold T, Heinrich R (2006) The DLR TAU-code: recent applications in research and industry. In: Proceedings of ECCOMAS CFD 2006. Delft University of Technology

Finite Transformation Rigid Motion Mesh Morpher



Athanasios G. Liatsikouras, Guillaume Pierrot,
Gabriel Fougeron and George S. Eleftheriou

Abstract In any optimization framework, a robust and reliable mesh morpher is necessary to undertake the adaptation of the CFD mesh to the updated boundaries at each optimization cycle. Morphing has its share of challenges, namely to maintain high mesh quality (avoid distorted elements and tangles) even during extreme deformations. In this work, the Finite Transformation Rigid Motion Mesh Morpher (FT-R3M) is presented, an improved version of the Rigid Motion Mesh Morpher (Eleftheriou and Pierrot in Rigid motion mesh morpher: a novel approach for mesh deformation, 2016), that eliminates the need for sub-cycling, making it more efficient in terms of CPU time. FT-R3M, which bears some similarities to Chal et al. (ACM Trans Graph 29(4):38, 2010), is a mesh-less mesh morphing tool, since it does not require any inertial quantities, that gracefully propagates the movement of the boundaries (surface mesh) to the internal nodes of the mesh (volume mesh), by keeping the motion of its parts (referred to as stencils) as-rigid-as-possible. It is an optimization-based method, which means that the interior nodes of the computational mesh are displaced to minimize a distortion metric, namely the deformation energy. Since FT-R3M is minimizing the deformation energy between the initial

A. G. Liatsikouras (✉)

ESI Software Germany GmbH, Kruppstr. 90, 45145 Essen, Germany
e-mail: athanasios.liatsikouras@esi-group.com; thanasis.liatsi@gmail.com

A. G. Liatsikouras

Parallel CFD & Optimization Unit, School of Mechanical Engineering,
National Technical University of Athens (NTUA), Athens, Greece

G. Pierrot · G. Fougeron · G. S. Eleftheriou

ESI-Group, 99 Rue des Solets, 94513 Rungis, France
e-mail: guillaume.pierrot@esi-group.com

G. Fougeron

e-mail: gabriel.fougeron@esi-group.com

G. S. Eleftheriou

e-mail: george.eleftheriou@esi-group.com

G. Fougeron

CentraleSupélec Laboratoire MSSMat, UMR CNRS 8579,
Grande Voie des Vignes, 92290 Châtenay-Malabry, France

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_7

and the final configuration, as opposed to R3M, in which the deformation energy is minimized from each sub-cycle to another, there is a significant gain in terms of the quality of the resulting mesh. The efficiency of the morpher proposed in this article will be demonstrated in small and medium-size cases.

Introduction

A variety of stochastic and gradient-based methods have been devised to solve aerodynamic shape optimization problems such as the design of airfoils or wings for optimal drag and/or lift, ducts for minimum power losses, cars with optimal combination of drag and lift etc. In these kind of optimization problems, it is required to use a technique so as to deal with the necessary changes of the boundaries (surface mesh), namely to adapt the computational mesh to the updated geometry in order to proceed with the optimization process.

One well-known method to handle the changes in the boundaries during optimization, is remeshing, in which after every optimization cycle a new grid is generated. This is a time-consuming process and especially for gradient-based methods, in which the information needed for the shape changes is retrieved from the gradient with respect to free design variables, the gradient consistency is lost. Moreover, for complex geometries, manual intervention in the mesh generation will be needed, which makes this method neither robust nor reliable. On the other hand, a promising way to propagate the movement of a shape to the interior mesh is to make use of a mesh deformation tool (mesh morpher). This requires to generate a mesh only once, at the beginning of the optimization process and then, the mesh morpher will undertake to deform this mesh at each optimization cycle. In this article, the Finite Transformation Rigid Motion Mesh Morpher will be presented and demonstrated.

The challenge when a mesh morpher is used, is to maintain a good mesh quality of the deformed mesh, namely to avoid negative or distorted elements, highly skewed cells etc, which may cause divergence of the CFD software, in CFD-based problems. In literature, there is a variety of mesh deformation techniques. Most of them require a trade-off between attained mesh quality and CPU cost. A simple and easy way, in terms of implementation complexity, to deform a mesh is by using the Laplacian smoothing (Hansbo 1995; Su et al. 2010), or the so-called Laplacian coordinates. The computation of the displacement related to the nodes of the CFD mesh requires the solution of a linear system whose dimension scales with the number of nodes of the mesh. This technique proves to be efficient but not very robust, since it cannot handle mesh rotations and generally complex transformations. A rival mesh deformation tool is the Linear Elasticity morpher, in which the computational mesh is handled as an elastic solid body (Lynch 1982; Stein et al. 2003). In this method, mesh deformation is accomplished by solving the linear elasticity equations for the mesh point inside the geometric domain. Since the elasticity equations contain material properties (Young's modulus and Poisson's ratio), these are related to the mesh characteristics. Another popular mesh deformation method is the Spring Analogy Method (Batina 1991). In

this approach, each edge of the mesh is ‘replaced’ by a tension spring with the spring stiffness as the inverse of the edge length. Unfortunately, this method does not prevent inverted elements and it fails when the local mesh motion exceeds significantly the local mesh size. Many improvements have been achieved to prevent element inversion by introducing torsional springs (Farhat et al. 1998) or semi-torsional springs (Blom 2000), but still mesh anisotropies cannot be handled. A very promising mesh deformation technique is the mesh morpher based on Radial Basis Functions (RBF) (Jakobsson and Amignon 2007). Mesh morphing based on RBF is a robust and reliable mesh deformation technique but computationally very “heavy”, as the matrices involved in the computations are very dense.

The Rigid Motion Mesh Morpher has already been introduced (Eleftheriou and Pierrot 2016). It has been shown that it can handle mesh rotation and mesh anisotropy very efficiently. An improvement upon this concept, is the development of the Finite Transformation Rigid Motion Mesh Morpher, that eliminates the need for sub-cycling and keeping track of the ‘rigid–motion’ history of the stencils for all sub-cycles. It employs, as it will be explained below, the Polar decomposition method in order to project an estimation of the rotation matrix to the special orthogonal space $SO(n)$ (rotation group; group of the orthogonal matrices of determinant 1), where n is the spatial dimensions of the problem. The basic idea of FT–R3M is to adapt/deform the nodes, whose displacement is not prescribed, to a given displacement field of the prescribed nodes (in most cases the prescribed nodes correspond to the boundary nodes of the shape), by keeping parts/elements of the mesh as–rigid–as–possible, hence keeping the deformation energy of these parts/elements, between the initial and the final state of the shape, minimal. There is a significant gain in terms of CPU and morphing efficiency.

Rigid Motion as a Building Block Towards Mesh Morphing

The Finite Transformation Rigid Motion Mesh Morpher is a minimization–based approach. There is a target functional, namely the deformation energy in our context, which has to be minimized in order to propagate the movement of the boundaries to the internal nodes of the CFD mesh. In the subsections that follow, the rigid motion and the total deformation energy to be minimized will be introduced.

Rigid Motion

A rigid motion (or an *isometry*) consists of rotations, translations (or a combination of them) such that the distance between every pair of points/vectors is preserved. In particular, a rigid motion is defined as a map $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ that conserves the inner product

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \quad \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \langle \mathbf{x}, \mathbf{y} \rangle \quad (1)$$

Assuming a rotation matrix $R \in SO(n)$ ($R \cdot R^T = I_n$, where I_n is the identity matrix) and a translation vector $\mathbf{t} \in \mathbb{R}^n$, a rigid motion when acting on any vector \mathbf{x} , produces the transformed vector $\phi(\mathbf{x})$ of the form

$$\phi(\mathbf{x}) = R\mathbf{x} + \mathbf{t} \quad (2)$$

The Deformation Energy

Let us denote by \mathbb{N} the set of nodes, $\partial\mathbb{N}$ the boundary nodes and $\tilde{\mathbb{N}}$ the internal nodes. From now on, let \mathbf{X}_i be the initial position vector of node with index i and \mathbf{x}_i the final one. If the boundary nodes follow a rigid body motion, we also want to a-priori impose to the interior nodes of the mesh the same rigid body motion

$$\begin{aligned} (\exists(R, \mathbf{t}) \in SO(n) \times \mathbb{R}^n \mid \forall i \in \partial\mathbb{N}, \mathbf{x}_i = R \mathbf{X}_i + \mathbf{t}) \Rightarrow \\ (\forall k \in \tilde{\mathbb{N}}, \mathbf{x}_k = R \mathbf{X}_k + \mathbf{t}) \end{aligned} \quad (3)$$

To this end, this leads to the following optimization problem: find the pair $(R, \mathbf{t}) \in SO(n) \times \mathbb{R}^n$ and $\mathbf{x}_i \in \tilde{\mathbb{N}}$ that minimizes the ‘energy functional’ (deformation energy) between the initial and the final state

$$\begin{aligned} E(R, \mathbf{t}, \mathbf{x}_1, \dots, \mathbf{x}_n) &= \sum_{i \in \mathbb{N}} \|\phi(\mathbf{X}_i) - \mathbf{x}_i\|^2 \\ &= \sum_{i \in \mathbb{N}} \|R \mathbf{X}_i + \mathbf{t} - \mathbf{x}_i\|^2 \end{aligned} \quad (4)$$

Equivalently, Eq. 4 could be written for the edges of the mesh instead of the nodes as follows

$$E = \sum_{(i,j) \in \mathbb{N}} \|R(\mathbf{X}_j - \mathbf{X}_i) - (\mathbf{x}_j - \mathbf{x}_i)\|^2 \quad (5)$$

By minimizing expression 5, the whole mesh is handled as one body. In order to make the method more flexible and robust, we group some parts/elements of the mesh into ‘stencils’. A stencil can be a collection of edges, not necessarily geometrical edges. In our context, an edge is a pair of nodes sharing a common cell. There is a freedom in the selection of the stencils; in the simplest case, a 1:1 ratio between the number of stencils and the number of elements is considered. Decomposing the final deformation energy as a weighted summation of the deformation energy of each stencil belonging to the mesh, the final expression of the deformation energy to be minimized can be written as

$$E = \sum_{s \in \mathbb{S}} w_s \sum_{(i,j) \in s} \mu_{s,ij} \|R_s(\mathbf{X}_j - \mathbf{X}_i) - (\mathbf{x}_j - \mathbf{x}_i)\|^2 \quad (6)$$

In Eq. 5, \mathbb{S} is the stencil set, s is a stencil belonging to this and $(i, j) \in s$ denotes an edge belonging to stencil s ; w_s is a positive scalar weight per stencil that stresses the importance of some stencils as higher than some others' (e.g. in case of a boundary layer) and, $\mu_{s,ij}$ a positive scalar weight per stencil–edge that accounts for mesh anisotropy, by preventing the distortion of a stencil by favouring rigidity in directions in which distortion is imminent. Over and above, by denoting as

$$\mathbf{e}_{s,ij} = \sqrt{w_s \mu_{s,ij}} \left[R_s(\mathbf{X}_j - \mathbf{X}_i) - (\mathbf{x}_j - \mathbf{x}_i) \right] \quad (7)$$

the total deformation energy in Eq. 5 to be minimized can be rewritten as

$$E = \sum_{s \in \mathbb{S}} \sum_{(i,j) \in s} \|\mathbf{e}_{s,ij}\|^2 \quad (8)$$

In the final expression of the deformation energy stated in Eq. 5, there is a hidden non–linear constraint; namely, every matrix R_s (one for each stencil of the mesh) should be a ‘real’ rotation matrix inside the orthogonal $SO(n)$ group, which can be expressed as

$$\forall s \in \mathbb{S}, \quad R_s^T R_s = I_n \quad (9)$$

Finally, the Energy to be minimized can be expressed as

$$\left\{ \begin{array}{l} \sum_{s \in \mathbb{S}} \sum_{(i,j) \in s} \|\mathbf{e}_{s,ij}\|^2 \\ s.t. \quad R_s^T R_s = I_n \end{array} \right. \quad (10)$$

Algorithm of the Morphing Framework

After having introduced the total deformation energy in Eq. 8, it is worth describing the algorithm to be followed in order to minimize it and, obtain the new/updated position of every internal node constituting the CFD mesh. Since the problem expressed in Eq. 8 is non–linear, it has to be linearized and follow an iterative process to achieve convergence. Initially, the quadratic constraint stated in Eq. 8 should be linearized. After that follows the derivation of the linearized deformation energy w.r.t. the unknowns, namely the nodal positions and a skew–symmetric matrix for each stencil (to be explained later on) and the assembly of a linear system to be solved. The necessary steps to be undertaken by the morpher are:

1. Solution of a linear system (to be explained later on) and computation of $(\mathbf{x}_i^{\kappa+1}, R_s^*)$ from $(\mathbf{x}_i^\kappa, R_s^\kappa)$, the nodal positions and an estimation of the rotation matrix respectively. Since this solution was obtained by linearizing the constraint in Eq. 8, R_s^* will not be a rotation matrix that belongs to the orthogonal group $SO(n)$, thus

it has to be projected back to $SO(n)$ in order to find the closest rotation matrix to R_s^* . This step is called *prediction step*.

2. Computation of the closest rotation matrix to R_s^* . In this step, the position vectors $\mathbf{x}_i^{\kappa+1}$ are kept fixed and, the R_s^* for every stencil are projected to the orthogonal group $SO(n)$ to obtain the closest rotation matrix $R_s^{\kappa+1}$. This step is called *correction step*.
3. Check if steady state has been achieved, namely if \mathbf{x}_i, R_s have converged. If not, the iterative process continues returning to step 1.

Building the System of Equations

Before going to the linearization of the total deformation energy, it is worth mentioning a few things about the structure of the orthogonal group and its tangent space, that will also be used during the linearization of the energy. Furthermore, in this section, the linearized energy will be constructed, as well as the necessary steps to build the final system to be solved whose solution is the position of each node of the mesh.

Structure of the Orthogonal Group and Its Tangent Space

To better understand the implicit quadratic constraint in Eq. 8, the structure of the orthogonal group and its tangent space will be briefly explained. Let us denote

$$SO(n) = \{R \in GL(n, \mathbb{R}) \mid R^T R = R R^T = I_n\} \quad (11)$$

where $GL(n, \mathbb{R})$ is the general linear group of degree n (the multiplication of two invertible matrices is also an invertible matrix). In addition, let us denote as

$$\varphi(R) = R^T R - I_n = 0_n \quad (12)$$

Since $\varphi(R) = 0_n$ is singleton (unit set; contains exactly one element), it is also a closed set. Moreover, $\varphi(R)$ is continuous, hence $SO(n)$ is a closed set. Let $R \in SO(n)$ and $B \in GL(n, \mathbb{R})$ be given and $\varepsilon \in \mathbb{R}$, then we have

$$\begin{aligned} \varphi(R + \varepsilon B) &= (R + \varepsilon B)^T (R + \varepsilon B) - I_n \\ &= R^T R - I_n + \varepsilon(B^T R + R^T B) + \varepsilon^2 B^T B \\ &= \varphi(R) + \varepsilon(B^T R + R^T B) + \varepsilon^2 B^T B \end{aligned} \quad (13)$$

Hence, the Gâteaux derivative of φ is

$$\begin{aligned} d\varphi(R, B) &= \lim_{\varepsilon \rightarrow 0} \frac{\varphi(R + \varepsilon B) - \varphi(R)}{\varepsilon} \\ &= \left. \frac{d}{d\varepsilon} \varphi(R + \varepsilon B) \right|_{\varepsilon=0} \\ &= B^T R + R^T B \end{aligned} \quad (14)$$

Since $d\varphi(R, B)$ is linear and continuous, it follows that $SO(n)$ is a differentiable manifold and that its tangent space at R ($T_R SO(n)$) is the null-space of $d\varphi(R, B)$:

$$T_R SO(n) = \{d\varphi(R, B) = B^T R + R^T B = 0_n\} \quad (15)$$

which means that the space tangent to the orthogonal group $SO(n)$ at R is the space of linear transformations B such that $R^T B$ is skew-symmetric.

Linearization of the Problem

Assuming that a current guess (\mathbf{x}_i^k, R_s^k) is available and by introducing the increments $(\delta \mathbf{x}_i, \delta R_s)$, the energy expressed in Eq. 8 (using also Eq. 7) can be rephrased as

$$\left\{ \begin{array}{l} \sum_{s \in \mathbb{S}} w_s \sum_{(i,j) \in s} \mu_{s,ij} \left[(R_s^k + \delta R_s)(\mathbf{X}_j - \mathbf{X}_i) \right. \\ \left. - (\delta \mathbf{x}_j - \delta \mathbf{x}_i) - (\mathbf{x}_j^k - \mathbf{x}_i^k) \right]^2 \\ s.t. \quad R_s^k \delta R_s^T + \delta R_s R_s^{kT} = -\delta R_s \delta R_s^T \end{array} \right. \quad (16)$$

Now, the linearization of the constraints takes place at this point, by neglecting the second order terms (assuming that $\delta R_s \delta R_s^T = 0$). Then, introducing $\Omega_s = \delta R_s R_s^T$, a skew-symmetric matrix (see also Section “Structure of the Orthogonal Group and Its Tangent Space”, Eq. 16 becomes

$$\left\{ \begin{array}{l} \sum_{s \in \mathbb{S}} w_s \sum_{(i,j) \in s} \mu_{s,ij} \left[(I_n + \Omega_s) R_s^k (\mathbf{X}_j - \mathbf{X}_i) \right. \\ \left. - (\delta \mathbf{x}_j - \delta \mathbf{x}_i) - (\mathbf{x}_j^k - \mathbf{x}_i^k) \right]^2 \\ s.t. \quad \Omega_s + \Omega_s^T = 0 \end{array} \right. \quad (17)$$

Finally, since Ω_s is antisymmetric, it implies that the product $\Omega_s [R_s^k (\mathbf{X}_j - \mathbf{X}_i)]$ can be replaced by $\mathbf{b}_s \times [R_s^k (\mathbf{X}_j - \mathbf{X}_i)]$ with \mathbf{b}_s a vector. By taking into consideration

the latter and by rearranging some terms in Eq. 17 we end up with

$$\left\{ \begin{array}{l} \sum_{s \in \mathbb{S}} w_s \sum_{(i,j) \in s} \mu_{s,ij} \left[\mathbf{b}_s \times [R_s^k(\mathbf{X}_j - \mathbf{X}_i)] \right. \\ \left. - (\delta \mathbf{x}_j - \delta \mathbf{x}_i) - (\mathbf{x}_j^k - \mathbf{x}_i^k) + R_s^k(\mathbf{X}_j - \mathbf{X}_i) \right]^2 \\ s.t. \quad \Omega_s + \Omega_s^T = 0 \end{array} \right. \quad (18)$$

Once the displacement of the prescribed nodes (namely the displacement of the nodes in the surface mesh) is known, the deformation energy in Eq. 18 is minimized in a least squares sense, by finding the stationary points w.r.t. the corresponding unknowns by satisfying

$$\frac{\partial E}{\partial \delta \mathbf{x}_i} = 0 \quad \text{and} \quad \frac{\partial E}{\partial \mathbf{b}_s} = 0 \quad (19)$$

The derivation of the linearized total deformation energy w.r.t. the degrees of freedom, namely $\delta \mathbf{x}$ and \mathbf{b}_s , as stated at Eq. 19, yields a symmetric positive definite system to be solved of the form

$$\mathbb{A} \mathbf{u} = \mathbf{t} \quad (20)$$

where \mathbf{u} is the vector that consists of the free variables. Using the *condensation* (Paz and Leigh 2001) technique, also known as *static condensation*, we can eliminate \mathbf{b}_s in terms of $\delta \mathbf{x}_i$ so as to decrease the size of the system.

Projection on the Orthogonal Group

The linearization of the constraint in Eq. 16 leads to obtain a matrix R_s^* which lies on the tangent space of the orthogonal group $SO(n)$. This practically means that R_s^* it is not a ‘real’ rotation matrix, thus $R_s^* R_s^{*T} \neq I_n$. The latter makes the projection of the estimated R_s^* in the orthogonal group, essential.

A way to deal with such a problem, is to solve the *Wahba’s problem* (Wahba 1965). The latter seeks for a rotation matrix between two coordinates systems from a set of weighted vector observations. Assuming that all \mathbf{x}_i are known, which in our case are the nodal position vectors computed after the minimization of Eq. 18, the rotation matrix R_s of stencil $s \in \mathbb{S}$ is computed by minimizing the stencil energy

$$\left\{ \begin{array}{l} W(R) = \sum_{(i,j) \in s} \mu_{s,ij} \left\| |R_s(\mathbf{X}_j - \mathbf{X}_i) - (\mathbf{x}_j - \mathbf{x}_i)| \right\|^2 \\ s.t. \quad R_s R_s^T = I_n \end{array} \right. \quad (21)$$

According to *Wahba's problem*, the minimization of Eq. 21 is related to the orthogonal Procrustes problem, which is to find the orthogonal matrix R_s that is closest to B , where $B = \sum_k \mu_{s,ij}(\mathbf{x}_j - \mathbf{x}_i)(\mathbf{X}_j - \mathbf{X}_i)^T$, in the sense of Frobenius norm

$$\|R_s - B\|_F^2 = \|R_s\|_F^2 + \|B\|_F^2 - 2tr(R_s B^T) \quad (22)$$

Since R_s is a-priori imposed to be an orthogonal matrix, with the proviso that the determinant is +1 (otherwise if the determinant is -1, it will still be orthogonal, but a reflection matrix), $tr(R_s R_s^T) = n$, where n is the dimension of the problem and matrix B is known, the term $-2tr(R_s B)$ in Eq. 22 should be minimized. Hence R_s is nothing more than the unitary polar factor of the matrix denoted as B .

In the literature there are many ways to compute the polar decomposition of a matrix (Markley 1988; Higham 1986). Herein, an iterative method is used based on *Heron's method* for the square root of 1, which has been proved to be very efficient.

Applications

In this section, the Finite Transformation Rigid Motion Mesh Morpher, proposed in this work, is being demonstrated and tested as a stand-alone tool to show its efficiency in three cases, in particular in the moving boxes, in a rotating airfoil and in a beam fixed at the wall.

Moving Boxes Case

This case deals with the movement of a box, which is placed inside a larger box. It is a simple case, at first sight, though it can provide useful information for the efficiency and the behaviour of the morpher proposed in this work.

The mesh in between the two boxes consists of 2400 hexahedral elements and approximately 5000 nodes. In the first case, the inner box is rotated 80 degrees anti-clockwise around its center and in the second one, it is translated in x-direction (by one time its box width). In both cases, the outer box is kept fixed (not deformable) and the nodes in between the two boxes are adapted to each movement. The initial mesh is presented in Fig. 1 whereas the deformed meshes for both cases, are presented in Fig. 2. The purpose in the first case is to test the FT-R3M in mesh rotation whereas in the second one, is to show how the proposed mesh morpher treats the squeezed elements.

In order to quantify the mesh quality, two quality metrics are used. Table 1 tabulates these quality metrics (maximum value of skewness and non-orthogonality metric) for each mesh (initial and deformed). For the sake of completeness, it is worth mentioning that the non-orthogonality metric measures the angle between the line

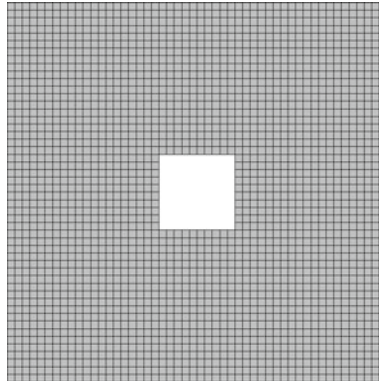


Fig. 1 Initial mesh in between the two boxes

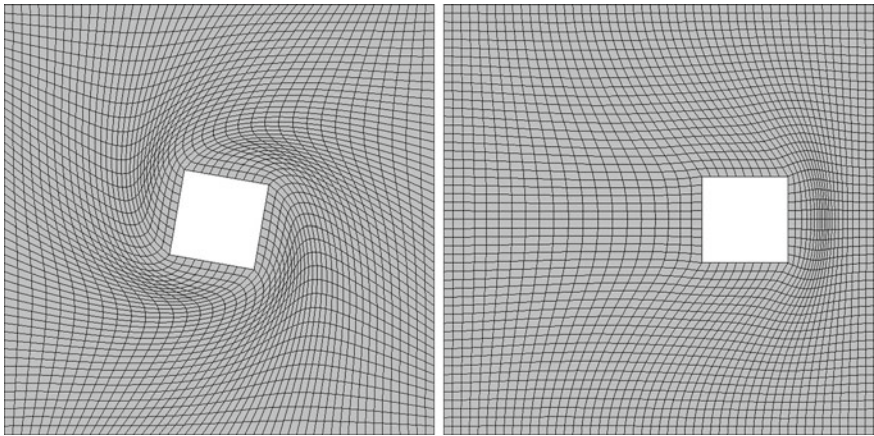


Fig. 2 Initial mesh in between the two boxes. In fig b the inner square has been rotated x degrees around its center and in fig c it is translated over the x -axis

Table 1 Non-Orthogonality and skewness metric before and after the movement of the inner box. Left: Quality metrics in the case of the rotation of the inner box. Right: Quality metrics in the case of the translation of the inner box

Quality metric	Max. value		Quality metric	Max. value	
	Before	After		Before	After
Non-Orthogonality	0°	66.18°	Non-Orthogonality	0°	35.87°
Skewness	0	0.91	Skewness	0	0.51

connecting two cell centers and the normal of their common face (the lower the better) whereas the skewness metric measures the distance between the intersection of the line connecting two cell centers with their common face and the center of that face (smaller is better).

In this simple case, it is illustrated that the Finite Transformation Rigid Motion Mesh Morpher is able to handle mesh rotations and, at the same time, maintain a good mesh quality in such extreme deformations. FT-R3M can also prevent the distortion of an element (squeezed elements) by keeping these stencils more rigid than others for which the distortion is not imminent.

Deformation of a 2D Airfoil

The second problem is dealing with the rotation of a 2D airfoil. The computational mesh is appropriate for viscous flows and consists of approximately 32K nodes and 60K elements, with a boundary layer at walls (quadrilateral elements) and triangular elements everywhere else. The airfoil is rotated anti-clockwise and FT-R3M undertakes the CFD mesh adaptation to the rotated boundaries. Moreover, quality metrics (in particular skewness and non-orthogonality) are used to quantify the quality of the resulting mesh. The purpose is to demonstrate the utility of the proposed mesh morpher in extreme deformations in an anisotropic mesh. In Fig. 3, the deformation energy on the initial and the deformed mesh of the airfoil are presented.

In Fig. 4 a focus has been made on the leading (left) and the trailing (right) edge of the airfoil on the resulting mesh, after 90° degrees of rotation for demonstration purposes. The skewness and the non-orthogonality metrics for the initial and the resulting mesh are tabulated in Table 2. The rotation of 90° degrees of the airfoil is just before an inverted element appears. In Fig. 4 it can be observed that FT-R3M preserves the orthogonality of the boundary layer on the deformed airfoil.

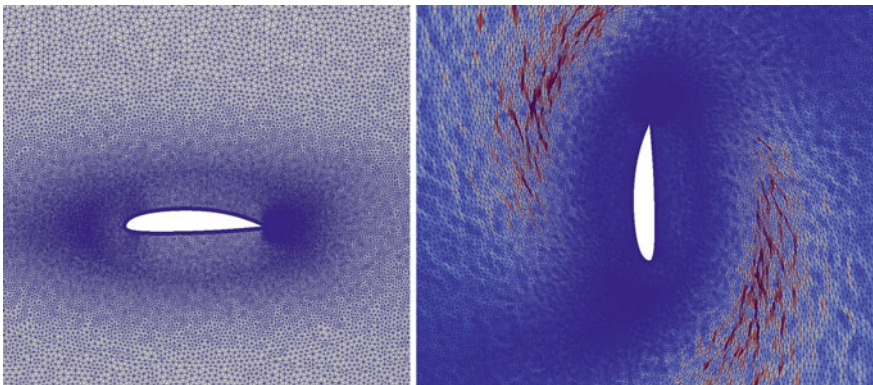


Fig. 3 Initial mesh on the airfoil (left) and the deformation energy on the deformed airfoil (right)

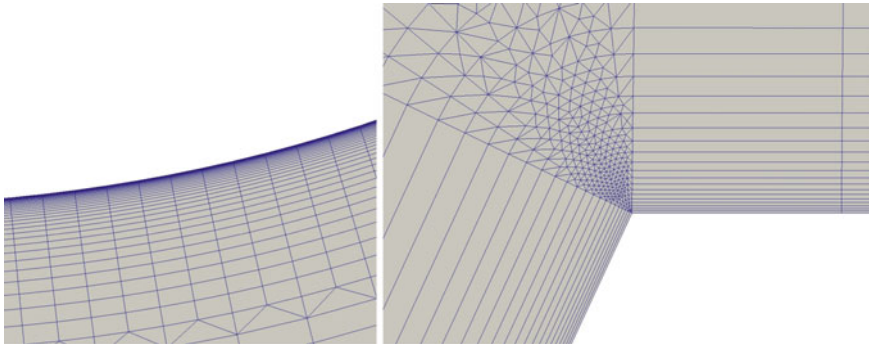


Fig. 4 Focus on the leading (left) and trailing (right) edge of the rotated airfoil

Table 2 Mesh quality metrics before and after the deformation of the airfoil

Quality metric	Max. value	
	Before	After
Non-Orthogonality	56.59°	84.21°
Skewness	0.88	0.90

Deformation of a Beam

The last case, in which FT-R3M is tested, deals with the deformation of a beam with a rectangular cross section which is fixed at a wall. The aim is to apply extreme deformations in the free cross-section and adapt the mesh in the new shape of the beam. The initial shape and mesh of the beam is demonstrated in Fig. 5; the mesh has been generated using CFD-GEOM and consists of approximately 195K elements (tetrahedra) and 79K nodes.

The deformed beam is demonstrated in Figs. 6 and 7 for different displacement fields (bending and twisting deformation respectively). In addition, the same quality metrics that have been used in previous chapters are also used here to monitor the quality of the deformed mesh and are tabulated in Table 3. Because of the extreme deformation field that is applied, the non-linear iterations needed in this case are more in number than these on the airfoil and the moving boxes cases.

In this particular case, the quality of the mesh is kept almost the same during the deformation (Table 3). FT-R3M is capable of handling extreme deformations and at the same time ensuring a good mesh quality.

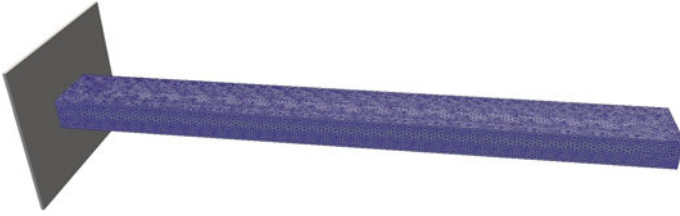


Fig. 5 Initial shape and mesh of the beam. The wall on the left of the figure appears only for demonstration purposes (It has no effect in the morphing process)

Fig. 6 Bending deformation of the beam. The free cross section attaches the wall in which the beam is fixed

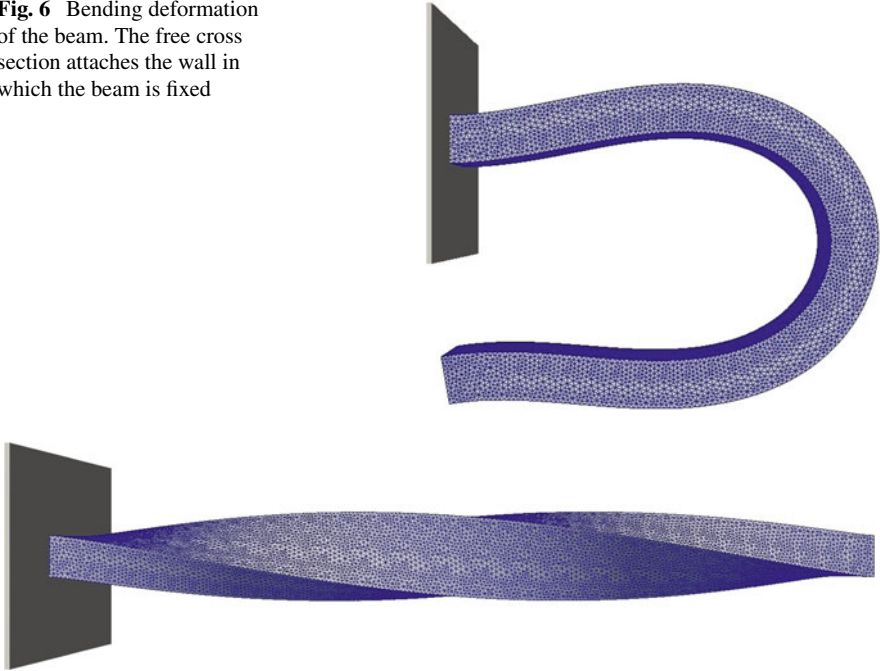


Fig. 7 The beam has been twisted 180° around the axis which is parallel to its length

Table 3 Quality metrics for the bending deformation (left) and the twisting deformation (right)

Quality metric	Max. value		Quality metric	Max. value	
	Before	After		Before	After
Non-Orthogonality	69.0912°	68.9014°	Non-Orthogonality	69.0912°	69.7431
Skewness	0.99047	0.99019	Skewness	0.99047	0.99037

Conclusion

In this work, a mesh deformation tool, the Finite Transformation Rigid Motion Mesh Morpher, has been developed and introduced. FT-R3M belongs to the family of optimization-based methods, since the internal nodes of the mesh are displaced to minimize a distortion metric. It has been demonstrated that FT-R3M can handle intrinsically mesh anisotropies and mesh rotations even in extreme deformations and maintains a good quality of the resulting mesh.

In this article, FT-R3M has been demonstrated as a stand-alone tool, thus as on-going work, it will be to couple it with ESI's in-house adjoint solver (Oriani and Pierrot 2016). To do so, the equations that describe the morpher should be differentiated so as to compute the adjoint counterpart of it and eventually to compute the grid sensitivities. Future work may include smoothing of the surfaces, which is necessary in an adjoint solver loop, since the adjoint sensitivity vector usually contains numerical noise. Moreover, the functionality of the FT-R3M may be extended in order to be used as static deformation tool, namely as a mesh optimization tool.

Acknowledgements This work has been conducted within the IODA ITN on: "Industrial Optimal Design using Adjoint CFD". The first author is an Early Stage Researcher (ESR) in this project which has received funding from the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Grant Agreement No. 642959.

References

- Batina J (1991) Unsteady Euler algorithms with unstructured dynamic mesh for complex-aircraft aerodynamic analysis. *AIAA J* 29(3):327–333
- Blom F (2000) Considerations on the spring analogy. *Int J Numer Methods Fluids* 32(6):647–668
- Chao I, Pinkall U, Sanan P, Schröder P (2010) A simple geometric model for elastic deformations. *ACM Trans Graph* 29(4):38
- Eleftheriou G, Pierrot G (2016) Rigid motion mesh morpher: a novel approach for mesh deformation. In: An international conference on engineering and applied sciences optimization, OPT-i, Kos, Greece
- Farhat C, Degant C, Koobus B, Lesoinne M (1998) Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Comput Methods Appl Mech Eng* 163(1–4):231–245
- Hansbo P (1995) Generalized Laplacian smoothing of unstructured grids. *Int J Numer Methods Biomed Eng* 11(5):455–464
- Higham NJ (1986) Computing the polar decomposition with applications. *SIAM J Sci Stat Comput* 7(4):1160–1174
- Jakobsson S, Amoignon O (2007) Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Comput Fluids* 36(6):1119–1136
- Lynch DR (1982) Unified approach to simulation on deforming elements with applications to phase change problems. *J Comput Phys* 47(3):387–411
- Markley F (1988) Attitude determination using vector observations and the singular value decomposition. *J Astronaut Sci* 36(3):245–258
- Oriani M, Pierrot G (2016) Alternative solution algorithms for primal and adjoint incompressible Navier-Stokes. In: ECCOMAS Congress 2016, VII European congress on computational methods in applied sciences and engineering, Crete Island, Greece

- Paz M, Leigh W (2001) Static Condensation and Substructuring. In: Integrated matrix analysis of structures, Elsevier, pp 239–260
- Stein K, Tezduyar T, Benney R (2003) Mesh moving techniques for fluid-structure interactions with large displacements. *J Appl Mech* 70(1):58–63
- Su Z, Wang S, Yu C, Liu F, Shi X (2010) A novel Laplacian based method for mesh deformation. *J Inf Comput Sci* 7(4):877–883
- Wahba G (1965) A least squares estimate of satellite attitude. *SIAM Rev* 7(3):409–409

The Unsteady Continuous Adjoint Method Assisted by the Proper Generalized Decomposition Method



V. S. Papageorgiou, K. D. Samouchos and Kyriakos Giannakoglou

Abstract In adjoint-based optimization for unsteady flows, the adjoint PDEs must be integrated backwards in time and, thus, the primal field solution should be available at each and every time-step. There are several ways to overcome the storage of the entire unsteady flow field which becomes prohibitive in large scale simulations. The most widely used technique is checkpointing that provides the adjoint solver with the exact primal field by storing the computed primal solution at a small number of time-steps and recomputing it for all other time-steps. Alternatively, approximations to the primal solution time-series can be built and used. One of them relies upon the use of the Proper Generalized Decomposition (PGD), as a means to approximate the time-series of the primal solution for use during the unsteady adjoint solver and this is where this paper is focusing on. The original contribution of this paper is that, apart from the standard PGD method, an incremental variant, running simultaneously with the time integration of unsteady primal equation(s) is proposed and tested. For the purpose of demonstration, three optimization problems based on different physical problems (unsteady heat conduction and unsteady flows around stationary and pitching isolated airfoils) are worked out by implementing the continuous adjoint method to both of them. The proposed incremental PGD technique is generic and can be used in any problem, to support either continuous or discrete unsteady adjoint.

Introduction

The numerical solution of the unsteady adjoint PDEs requires the storage or recomputation of the time-varying primal solution at each time-step. In the literature, strategies to overcome the full storage of the primal solution time-series have been

V. S. Papageorgiou · K. D. Samouchos · K. Giannakoglou (✉)
Parallel CFD & Optimization Unit, School of Mechanical Engineering,
National Technical University of Athens (NTUA), Athens, Greece
e-mail: kgianna@central.ntua.gr

K. D. Samouchos
e-mail: ksamouchos@yahoo.com

proposed. The most frequently used is (binomial) checkpointing (Griewank and Walther 2000). This may ensure the user-defined balance between storage of the primal solution at selected time-steps and recomputations. A viable alternative is to approximate the primal solution time-series through an (incremental) method with low storage requirements, employed during the solution of the primal PDEs, which are integrated forwards in time. Approximation methods can be evaluated in terms of the accuracy of the reconstructed primal solution and its effect on the computed sensitivity derivatives, as well their computational cost. Approximation methods have the advantage of avoiding (even partial) recomputations of the unsteady primal solution, as required by methods such as checkpointing. Among them, linear interpolation, cubic-splines, Fourier series or data compression techniques such as the Singular Value Decomposition (SVD) (Balzano and Wright 2013; Vezyris et al. 2016) should be reported.

In this paper, the PGD method (Chinesta et al. 2014; Ammar et al. 2012; Ladevèze 2014) is used to reconstruct the primal field by reducing memory storage. The main idea is to represent a multi-dimensional (in space and time) field as the sum of products of 1D functions; for an unsteady 2D scalar field of u , for instance, one may write

$$u(x, y, t) \cong \sum_{\mu=1}^M \phi^{\mu}(x)\theta^{\mu}(y)\tau^{\mu}(t) \quad (1)$$

Assuming that a small number M of modes is enough, a noticeable gain in memory usage is expected since scalar modes ϕ^{μ} , θ^{μ} and τ^{μ} ($\mu = 1, \dots, M$) are stored instead of the entire $u(x, y, t)$ field.

In an unsteady simulation, if the whole time-series of the solution $u(x, y, t)$ must be available before processing them by the PGD, no gain in storage requirements is expected. For this reason, an alternative method is proposed, in which once the instantaneous primal solution becomes available at each time-step, the already computed modes are incrementally updated. This will be referred to as **incremental PGD (iPGD)**.

In this paper, the programmed PGD (or iPGD) library is used to reconstruct the solution of an unsteady heat conduction and two unsteady inviscid flow problems within optimization workflows supported by the continuous adjoint method (Papadimitriou and Giannakoglou 2007). For the heat conduction problem, a 2D structured mesh is used and the spatial part of the space-time decomposition is performed in the transformed domain. In the unsteady flow problem, an immersed boundary approach [in specific, the cut-cell method (Clarke et al. 1986; Samouchos et al. 2016)] is used. The adaptive mesh used by the cut-cell method requires extra treatment that will be made clear in Section “[Applications 2 & 3: Gradient Computation for the Unsteady Euler Equations with the Cut-Cell Method](#)”.

Reconstructing Already Computed Fields by PGD

Consider a 2D time-dependent field $u = u(x, y, t)$, previously computed by any PDE solver on a standard structured mesh. In the PGD framework, this solution can be approximated by the sum of a relatively small number (M) of 1D functional products (Eq. 1). All these modes are built in M successive steps. At the m th step ($m \leq M$), the corresponding 1D functions are computed so as to minimize the representation error, which is given in discrete form by:

$$E_m = \frac{1}{2} \sum_{k=1}^K \sum_{i=1}^I \sum_{j=1}^J \left[\sum_{\mu=1}^m \phi_i^\mu \theta_j^\mu \tau_k^\mu - u_{i,j,k} \right]^2 \quad (2)$$

The problem of defining the modes is non-linear, so it has to be solved iteratively within each of the M steps of the successive enrichment by means of an alternating direction scheme. For the m th modes ($1 \leq m \leq M$), ϕ^m is computed first, considering θ^m and τ^m to be known from the previous iterations or their initialization and so forth. For instance to compute ϕ^m , Eq. 1, truncated by keeping only the first m terms, is multiplied by τ^m and θ^m and integrated along t and y . Since all the functions of y and t are known, the 2D integrals can be computed and the final equations for updating the modes are

$$\begin{aligned} \phi^m &= \frac{\int_y \int_t u \theta^m t^m dt dy - \sum_{\mu=1}^{m-1} \phi^\mu \int_y \int_t \theta^m \theta^\mu \tau^m \tau^\mu dt dy}{\int_y \int_t (\theta^m)^2 (\tau^m)^2 dt dy} \\ \theta^n &= \frac{\int_x \int_t u \phi^n t^n dt dx - \sum_{i=1}^{n-1} \theta^i \int_x \int_t \phi^n \phi^i \tau^n \tau^i dt dx}{\int_x \int_t (\phi^n)^2 (\tau^n)^2 dt dx} \\ \tau^n &= \frac{\int_y \int_x u \phi^n \theta^n dx dy - \sum_{i=1}^{n-1} \tau^i \int_y \int_x \phi^n \phi^i \theta^n \theta^i dx dy}{\int_y \int_x (\phi^n)^2 (\theta^n)^2 dx dy} \end{aligned} \quad (3)$$

The three above equations for the m th modes are used iteratively until an appropriate convergence criterion be met, before proceeding to the computation of the next modes.

Through differentiation of Eq. 2, it can be proved that the modes computed by Eqs. 3 minimize E_m . Taking this into consideration, the incremental variant of PGD (iPGD) can be formulated. This formulation and implementation of iPGD in unsteady adjoint is the key originality of this paper.

Incremental PGD Method

In order to approximate a flow field $u(x, y, t)$ using the method developed in the previous section, the whole time-series should have been computed and stored beforehand. This storage should definitely be avoided. This section presents a new method (iPGD) which overcomes this drawback. The concept of the iPGD method is that, the field reconstruction is gradually performed. During the integration of the primal PDEs, the solution field at each new time-step is used to enrich the previously computed modes.

Consider the same 2D time-dependent field $u = u(x, y, t)$ which, hereafter, will be in discrete form as $u_{i,j,k}$. The field approximation is still given by Eq. 1, but modes should incrementally be updated at each time-step. Equations for updating the modes are extracted by minimizing an error function similar to Eq. 2. Since only the current (time-index $k = K + 1$) solution field is available, the error function must be decomposed as:

$$E_m = \frac{1}{2} \sum_{i=1}^I \sum_{j=1}^J \left[\sum_{\mu=1}^m \phi_i^\mu \theta_j^\mu \tau_{K+1}^\mu - u_{i,j,K+1} \right]^2 + \frac{w}{2} \sum_{k=1}^K \sum_{i=1}^I \sum_{j=1}^J \left[\sum_{\mu=1}^m \left(\phi_i^\mu \theta_j^\mu \tau_k^\mu - \tilde{\phi}_i^\mu \tilde{\theta}_j^\mu \tilde{\tau}_k^\mu \right) \right]^2 \quad (4)$$

where the first term on the r.h.s. corresponds to the approximation error at the current time-step, whereas the second one to the overall error for all the previous time-steps, which have already been processed through the iPGD and yielded modes $\tilde{\phi}_i^\mu$, $\tilde{\theta}_j^\mu$, $\tilde{\tau}_k^\mu$. The contribution to the error is weighted by w which is user-defined. At each time-step, modes $(\phi_i^m, \theta_j^m, \tau_k^m)$ are updated and new values τ_{K+1}^m are appended. The unknown quantities are calculated by setting the derivatives of the error against zero, getting

$$\phi_i^m = Q_{1x}^i / Q_{2x}^i, \quad i = 1, \dots, I \quad (5a)$$

$$\theta_j^m = Q_{1y}^j / Q_{2y}^j, \quad j = 1, \dots, J \quad (5b)$$

$$\tau_k^m = Q_{1t}^k / Q_{2t}^k, \quad k = 1, \dots, K \quad (5c)$$

$$\tau_{K+1}^m = Q_{1T}^{K+1} / Q_{2t}^{K+1}, \quad (5d)$$

where

$$Q_{1x}^i = \tau_{K+1}^m \sum_{j=1}^J \theta_j^m u_{i,j,K+1} - \tau_{K+1}^m \sum_{\mu=1}^{m-1} \left[\left(\sum_{j=1}^J (\theta_j^\mu \theta_j^m) \right) \phi_i^\mu \tau_{K+1}^\mu \right] \\ + w \tilde{\phi}_i^m \sum_{k=1}^K \sum_{j=1}^J \tilde{\theta}_j^m \theta_j^m \tilde{\tau}_k^m \tau_k^m - w \sum_{k=1}^K \sum_{j=1}^J \left[\sum_{\mu=1}^{m-1} (\phi_i^\mu \theta_j^\mu \tau_k^\mu - \tilde{\phi}_i^\mu \tilde{\theta}_j^\mu \tilde{\tau}_k^\mu) \theta_j^m \tau_k^m \right]$$

$$Q_{2x}^i = (\tau_{K+1}^m)^2 \sum_{j=1}^J (\theta_j^m)^2 + w \sum_{k=1}^K \sum_{j=1}^J (\theta_j^m)^2 (\tau_k^m)^2$$

$$Q_{1y}^j = \tau_{K+1}^m \sum_{i=1}^I \phi_i^m u_{i,j,K+1} - \tau_{K+1}^m \sum_{\mu=1}^{m-1} \left[\left(\sum_{i=1}^I (\phi_i^\mu \phi_i^m) \right) \theta_j^\mu \tau_{K+1}^\mu \right] \\ + w \tilde{\theta}_j^m \sum_{k=1}^K \sum_{i=1}^I \tilde{\phi}_i^m \phi_i^m \tilde{\tau}_k^m \tau_k^m - w \sum_{k=1}^K \sum_{i=1}^I \left[\sum_{\mu=1}^{m-1} (\phi_i^\mu \theta_j^\mu \tau_k^\mu - \tilde{\phi}_i^\mu \tilde{\theta}_j^\mu \tilde{\tau}_k^\mu) \phi_i^m \tau_k^m \right]$$

$$Q_{2y}^j = (\tau_{K+1}^m)^2 \sum_{i=1}^I (\phi_i^m)^2 + w \sum_{k=1}^K \sum_{i=1}^I (\phi_i^m)^2 (\tau_k^m)^2$$

$$Q_{1t}^k = \tilde{\tau}_k^m \sum_{j=1}^J \sum_{i=1}^I \tilde{\phi}_i^m \phi_i^m \tilde{\theta}_j^m \theta_j^m - \sum_{i=1}^I \sum_{j=1}^J \sum_{\mu=1}^{m-1} (\phi_i^\mu \theta_j^\mu \tau_k^\mu - \tilde{\phi}_i^\mu \tilde{\theta}_j^\mu \tilde{\tau}_k^\mu) \phi_i^m \theta_j^m$$

$$Q_{1T}^{K+1} = \tau_{K+1}^m = \sum_{i=1}^I \sum_{j=1}^J \phi_i^m \theta_j^m u_{i,j,K+1} - \sum_{\mu=1}^{m-1} \left[\tau_{K+1}^\mu \sum_{i=1}^I \sum_{j=1}^J \phi_i^\mu \phi_i^m \theta_j^m \theta_j^\mu \right]$$

$$Q_{2t}^k = \sum_{i=1}^I \sum_{j=1}^J (\phi_i^m)^2 (\theta_j^m)^2$$

Equations 5 are coupled and must be solved iteratively through the following algorithm:

- **Step 1:** Initialize ϕ , θ and τ for $u(x,y,t)$ at the initial time instant, i.e. for $k=1$. This is, practically, equivalent to the PGD of a known 2D spatial field. Set $k=2$.
- **Step 2:** Compute all ϕ^m , θ^m and τ^m , $m=1, \dots, M$, using Eqs. 5a–5c.
- **Step 3:** Compute τ_{k+1}^m , using Eq. 5d.
- **Step 4:** Set $k=k+1$. Return to step 2.

It is obvious that the above algorithm approximates the various instantaneous spatial fields with different error as it proceeds from one time-step to the next. However, this cannot be avoided since we are using a single error function E which amalgamates all time instants; recall that the final error to be minimized in the one corresponding to all time-steps.

Application 1: Optimization Based on the Unsteady Heat Conduction Equation

The first application is dealing with the optimization of the temperature (T) profile along the left–most straight boundary S_c of a 2D domain Ω (Fig. 2). A 100×80 structured mesh is used. Along the remaining boundaries of Ω , fixed Dirichlet conditions are imposed on T . Temperature T (in Kelvin) along (S_c) is given by

$$T(\zeta, t) = \tilde{T}(\zeta) + 20\zeta(1 - \zeta) \sin\left(\frac{2\pi t}{T_a}\right) \quad (6)$$

where its “steady” part is

$$\begin{aligned} \tilde{T}(\zeta) = & \beta_1(5\zeta - 20\zeta^2 + 30\zeta^3 - 20\zeta^4 + 5\zeta^5) + \beta_2(10\zeta^2 - 30\zeta^3 + 30\zeta^4 - 10\zeta^5) \\ & + \beta_3(10\zeta^3 - 20\zeta^4 + 10\zeta^5) + \beta_4(5\zeta^4 - 5\zeta^5) \\ & + T_1(-5\zeta + 10\zeta^2 - 10\zeta^3 + 5\zeta^4 - \zeta^5) + T_2\zeta^5 \end{aligned}$$

This rather complicated expression results from a Bézier-based parameterization of the unknown temperature profile. In the above formulas, $0 \leq \zeta \leq 1$ is the non-dimensional distance of any node on S_c measured from the bottom-left corner of Ω , and $\beta_1, \beta_2, \beta_4$ are given by

$$\begin{aligned} \beta_1 = & \frac{1}{(b_1 - b_2)^2 + 3} - \frac{1}{(b_3 + 2)^2 + 5}, \quad \beta_2 = \frac{1}{(b_2 + b_3)^2 + 4} - \frac{1}{(b_3 - 1)^2 + 1}, \\ \beta_4 = & \frac{1}{(b_3 + 1)^2 + 2} - \frac{1}{(b_1 - b_3)^2 + 5} \end{aligned}$$

where b_q ($q=1, 2, 3$) are the three optimization variables. The extra variable β_3 depends on the other three, its role being to ensure that the mean temperature on the boundary S_c remains constant and equal to $450K$. This leads to the constraint $\beta_3 = 1800 - \beta_1 - \beta_2 - \beta_4$ to be met. The T profile in Eq. 6 changes periodically in time, with period $T_a = 800$ s. The unsteady heat conduction equation in the transformed (ξ, η) domain is

$$R = \rho C_p \frac{\partial T}{\partial t} - \frac{1}{J} \frac{\partial}{\partial \xi^i} \left(k J g^{ij} \frac{\partial T}{\partial \xi^j} \right) = 0 \quad (7)$$

where g^{ij} is the contravariant metrics and J the Jacobian of the transformation. The approximated temperature T , Eq. 1, in the (ξ, η) domain, takes the form $T(\xi, \eta, t) \cong \sum_{\mu=1}^M \phi^\mu(\xi) \theta^\mu(\eta) \tau^\mu(t)$. In Eq. 7, $k = k(T)$ is the thermal diffusivity, ρ the material density and C_p the heat capacity. Assuming that the material is aluminium, $k(T) = 0.0002213 T^2 - 0.09592 T + 211.5 [W/mK]$ (T in Kelvin), $\rho = 2.7 kg/m^3$ and $C_p = 0.910 kJ/kgK$. The length of boundary S_c is equal to 2m. The objec-

tive is to minimize the area and time with $T > T_{crit} = 400K$. Since this objective is not differentiable, it was replaced by

$$F = \frac{1}{\Omega T_a} \int_{t^*}^{t^*+T_a} \int_{\Omega} \left(1 - \frac{1}{1 + e^{k_2(T-T_{crit})+k_1}} \right) (aT + b) d\Omega dt \quad (8)$$

where $k_1 = 3$, $k_2 = k_1 / (T_{safe} - T_{crit})$, $a = 3 / T_{crit}$, $b = 1 - aT_{safe}$ and $T_{safe} = 450K > T_{crit}$ (a user defined relaxing threshold temperature). The time integral is extended over T_a with the lower limit of integration being t^* , at which the periodic solution has been established. The development of the continuous unsteady adjoint method is carried out in the standard way, leading to the field adjoint equation

$$-\rho C_p \frac{\partial \Psi}{\partial t} - \frac{1}{J} \frac{\partial}{\partial \xi^i} \left(k J g^{ij} \frac{\partial \Psi}{\partial \xi^j} \right) + g^{ij} \frac{\partial k}{\partial T} \frac{\partial T}{\partial \xi^i} \frac{\partial \Psi}{\partial \xi^j} = f \quad (9)$$

where Ψ is the adjoint field and f results from the differentiation of the objective function. Equation 9 is solved by imposing time periodic conditions $\Psi|_{t=t^*} = \Psi|_{t=t^*+T_a}$ whereas along the whole boundary S , $\Psi|_{S, \nu t} = 0$. Finally, the sensitivity derivatives of F w.r.t. the design variables b_q are given by

$$\frac{\delta F}{\delta b_q} = \int_{t^*}^{t^*+T_a} \int_{S_c} k \frac{\partial \Psi}{\partial n} \frac{\delta T}{\delta b_q} dS dt, \quad q = 1, 2, 3 \quad (10)$$

where $\frac{\delta T}{\delta b_q}|_{S_c} = \frac{\delta T}{\delta \beta_k} \frac{\delta \beta_k}{\delta b_q}$ ($k = 1, 2, 3, 4$). The optimization was based on the steepest descent method. The case ran for about 7 periods in order to ensure that the temperature field becomes periodic; then, the next period was processed by the proposed iPGD algorithm and stored for use during the integration of the unsteady adjoint equation in reverse time. 100 time-steps per period T_a were used. For the purpose of comparison, the same optimization was repeated twice: (a) with full storage of the results of the primal equations and (b) using the iPGD method for storing just the $\phi(\xi)$, $\theta(\eta)$ and $\tau(t)$ modes with various values of M . Of course, the full storage could have been replaced by checkpointing with identical sensitivity derivatives. In all cases, $w = 50$.

The sensitivity derivatives computed with fully stored and iPGD'ed primal fields are shown in Table 1. Reasonable deviations due to the approximation were expected but these were harmless for the optimization itself. In Fig. 2, initial and optimized time-averaged temperature fields are shown. The small overheated spot formed closed to S_c is not contradictory since a greater part of the area is kept at lower temperature and this yields lower values of F , Eq. 8. In Fig. 1a, the corresponding \tilde{T} distributions are shown. The optimization follows a slightly different path in each case (Fig. 1b), as the adjoint solver relies upon differently approximated temperature fields. However, all cases converge to close data-sets of the design variables by equally reducing

Table 1 Application 1: Sensitivity derivatives corresponding to the design variable data-set $b_1 = 0.05$, $b_2 = b_3 = 0.01$ for full storage of the primal field PGD and iPGD-based compression with various number of modes. Non-incremental a posteriori PGD (denoted by PGD $M = 20$) is also included in the plot

	$\delta F/\delta b_1$	$\delta F/\delta b_2$	$\delta F/\delta b_3$
Full Storage	-1.3715605463E-4	4.5884641458E-6	1.359568712E-2
PGD $M = 20$	-1.3749363010E-4	4.6796390425E-6	1.362148509E-2
iPGD $M = 30$	-1.4020728461E-4	4.3113618801E-6	1.391765592E-2
iPGD $M = 25$	-1.4067134121E-4	6.4219368409E-6	1.381274791E-2
iPGD $M = 20$	-1.4357302527E-4	6.7785627830E-6	1.406576690E-2

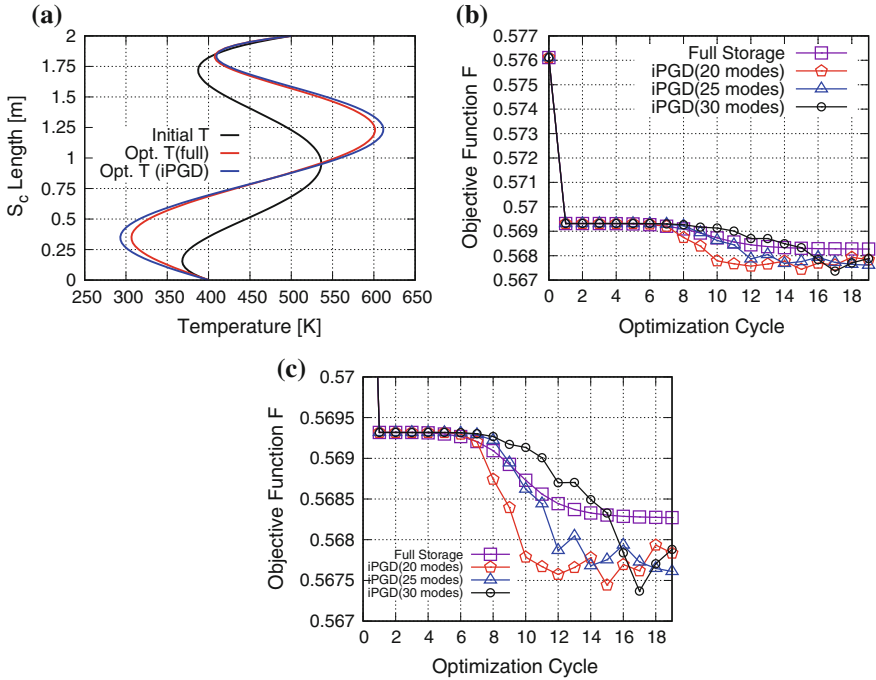


Fig. 1 Application 1: **a** Initial and optimal temperature profiles $\tilde{T}(\zeta)$ along S_c for full storage of the unsteady temperature field and using iPGD with $M = 30$. **b** Convergence of the objective function. **c** Blow-up view of (b)

the objective function. The memory needed, even with 30 modes, is about 32.5 % less than the full storage of the primal unsteady field. In terms of memory usage , we should make clear that the comparison is made against full storage, instead of checkpointing, to avoid also accounting for the extra computational cost of the latter due to the partial recomputation of the primal field.

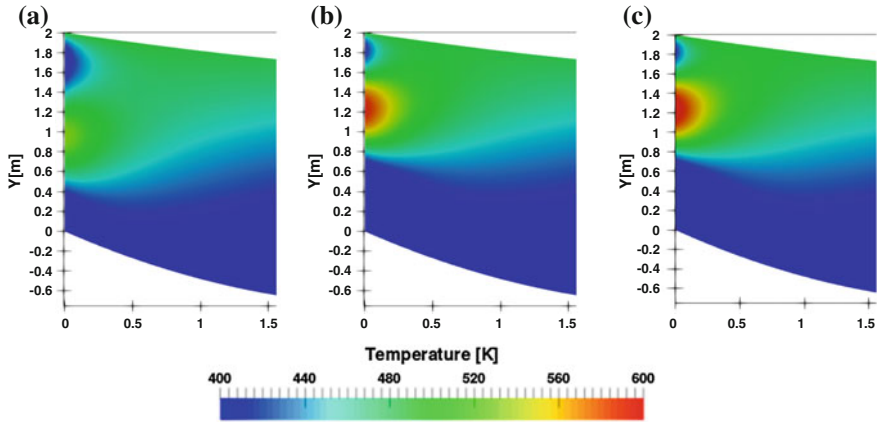


Fig. 2 Application 1: Time-averaged spatial T distribution on the left–most part of the computational domain for: **a** the initial \bar{T} profile (Eq. 8.4), with $b_1 = b_2 = b_3 = 0.01$, **b** the optimal one ($b_1 = 0.6903$, $b_2 = -0.3743$, $b_3 = -13.7670$) as computed with full storage of T and **c** the optimal solution ($b_1 = 0.8287$, $b_2 = -0.5367$, $b_3 = -14.0181$) computed using the iPGD with $M = 30$. Either optimization terminated after 20 cycles, using the same step of steepest descent

Applications 2 & 3: Gradient Computation for the Unsteady Euler Equations with the Cut-Cell Method

Herein, the PGD method is implemented to compute the objective function gradient to be used during the shape optimization of an isolated airfoil parameterized using Bézier curves, where the design variables (b_q) are their control points' coordinates. Gradient computation for a stationary airfoil, in which unsteadiness is introduced by the time-varying far-field flow angle and a pitching airfoil are demonstrated. In either case, the governing PDEs are the unsteady 2D flow Euler equations:

$$R_i = \frac{\partial U_i}{\partial t} + \frac{\partial f_{ij}}{\partial x_j} = 0, \quad i = 1, 4, \quad x_j = x, y \quad (11)$$

where $\vec{U} = [\rho \quad \rho u \quad \rho v \quad E]^T$ is the vector of conservative variables and f_{ij} are the inviscid fluxes in the Cartesian directions, ρ the fluid density, u and v the Cartesian velocity components, p the static pressure and E the total energy per unit volume. The objective function is the time-averaged lift over a single period, $F = \frac{1}{T_a} \int_0^{T_a} \int_{S_w} p n_k r_k dS dt$, where n_k , r_k are the components of the unit vectors normal to the airfoil surface and the freestream velocity, respectively.

The required derivatives of F w.r.t. b_q are computed by the continuous adjoint method. The adjoint equations are (Samouchos et al. 2016)

$$-\frac{\partial \Psi_i}{\partial t} - A_{jik} \frac{\partial \Psi_j}{\partial x_k} = 0 \quad i, j = 1, 4, x_k = x, y \quad (12)$$

where $A_{ijk} = \frac{\partial f_{ik}}{\partial U_j}$ and Ψ_i are the adjoint variable fields. The adjoint boundary conditions are omitted in the interest of space.

When the adjoint solution becomes periodic, the sensitivity derivatives are

$$\begin{aligned} \frac{\delta F}{\delta b_q} &= \frac{1}{T_a} \int_0^{T_a} \int_{S_w} p \frac{\delta(n_k r_k dS)}{\delta b_q} dt + \int_0^{T_a} \int_{S_w} (\Psi_{k+1} p - \Psi_i f_{ik}) \frac{\delta n_k}{\delta b_q} dS dt \\ &+ \int_0^{T_a} \int_{S_w} \Psi_i \left(\frac{\partial f_{il}}{\partial x_l} \frac{\delta x_k}{\delta b_q} - \frac{\partial f_{ik}}{\partial x_l} \frac{\delta x_l}{\delta b_q} \right) n_k dS dt \\ &+ \int_0^{T_a} \int_{S_w} \Psi_i \frac{\partial U_i}{\partial x_l} u_n^w \frac{\delta x_l}{\delta b_q} dS dt + \int_0^{T_a} \int_{S_w} (\Psi_i U_i + p \Psi_4) \frac{\delta u_n^w}{\delta b_q} dS dt \end{aligned}$$

The last two integrals are related to the normal velocity of the solid wall (u_n^w) and vanish for the stationary airfoil.

The flow solution is obtained using the cut-cell method (Clarke et al. 1986; Samouchos et al. 2016), according to which the flow simulation takes place on a Cartesian grid, covering both the fluid and solid regions. For higher accuracy, cells cut by the body contour along with their closest neighbours are subdivided into smaller ones (Fig. 3). Consequently, the (i, j) data-structure, being a prerequisite for applying the PGD, is no more valid. It is beyond the scope of this paper to compare the accuracy of the cut-cell method with that of CFD on body-fitted grids. Here, we are exclusively interested in evaluating the adequacy of the (i)PGD approximations.

The integration of the flow equations is based on a cell-centered finite-volume method with second-order accuracy in both space and time; fluxes are computed using the Roe scheme (Roe 1981). Special treatment for cells cut by the airfoil is needed in order to satisfy the conservation laws near solid boundaries with the required accuracy. An extra difficulty appears, if the body is moving in time. In such a case, the mesh is continuously adapted to the new position of the body, as follows. Firstly, the mesh undergoes a coarsening process, where all cells close to the solid wall are amalgamated with bigger neighbouring cells. Then, the body moves to its new position and, starting from the already coarsened mesh, cells are split until a certain level of refinement be met. Meshes are shown in Fig. 3.

By definition, the PGD (or iPGD) can be applied only to structured meshes. The lack of structure of the mesh used in the cut-cell method is overcome through a Reference/Regular (RR) mesh. After solving the unsteady Euler equations with the cut-cell method, the corresponding flow field is transferred to the RR mesh (Fig. 3c, d), which is as fine as the smallest cell of the cut-cell mesh (Fig. 3a, b). After that, the iPGD algorithm is implemented to the RR mesh as explained in

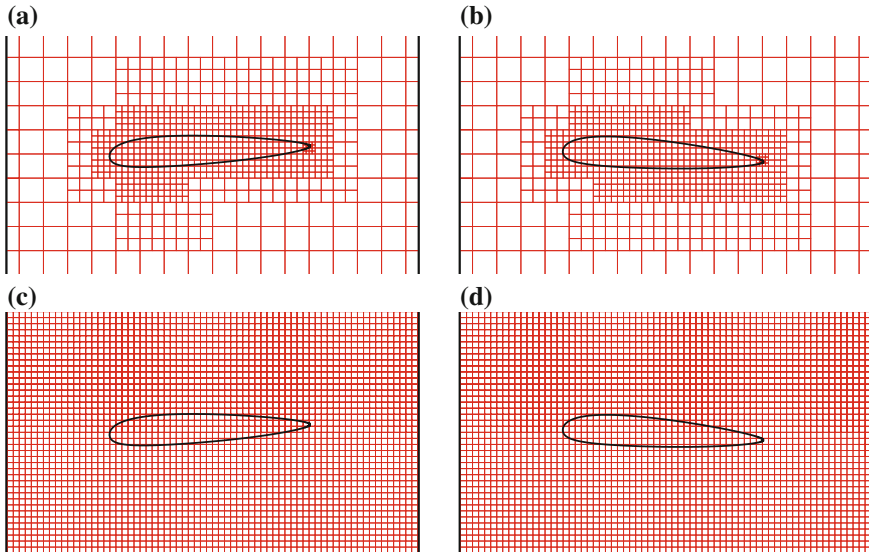


Fig. 3 Application 2.2: Adapted meshes, **a** and **b**, at the two extreme positions of the pitching airfoil. Corresponding RR meshes, **c** and **d**, used for the iPGD

Section “[Reconstructing Already Computed Fields by PGD](#)”. The required transition from the adapted cut-cell meshes to the RR one, must be quick and accurate. Using an efficient algorithm based on quad-tree data structure, the correspondence between the cells of the two meshes is easily accomplished. Each cell of the RR mesh takes on the flow variables of the cut-cell mesh cell which is part of.

Application 2: Stationary Airfoil

The unsteady Euler equations are solved around a stationary airfoil. The far-field flow conditions are $M_\infty = 0.3$ and $a_\infty = A \sin(\omega t)$ [deg] with amplitude $A = 3^\circ$ and period $T_a = \frac{2\pi}{\omega} = 0.015$ s. The cut-cell mesh used for the simulation consists of 10500 cells; a constant time-step equal to $T_a/20$ is used.

While integrating the flow equations from the previous to the current time-step, each flow field is processed by the iPGD with $M = 10$ and $w = 1000$. Figure 4 illustrates the pressure fields, at two instants corresponding to the max. and min. angle of attack, as computed by the cut-cell method. The comparison between approximated and exact fields is satisfactory.

After the time-integration of the flow equations, the iPGD’ed fields are made available to the adjoint software. A comparison of the computed adjoint energy field based on the exact and reconstructed flow solutions is presented in Fig. 5.

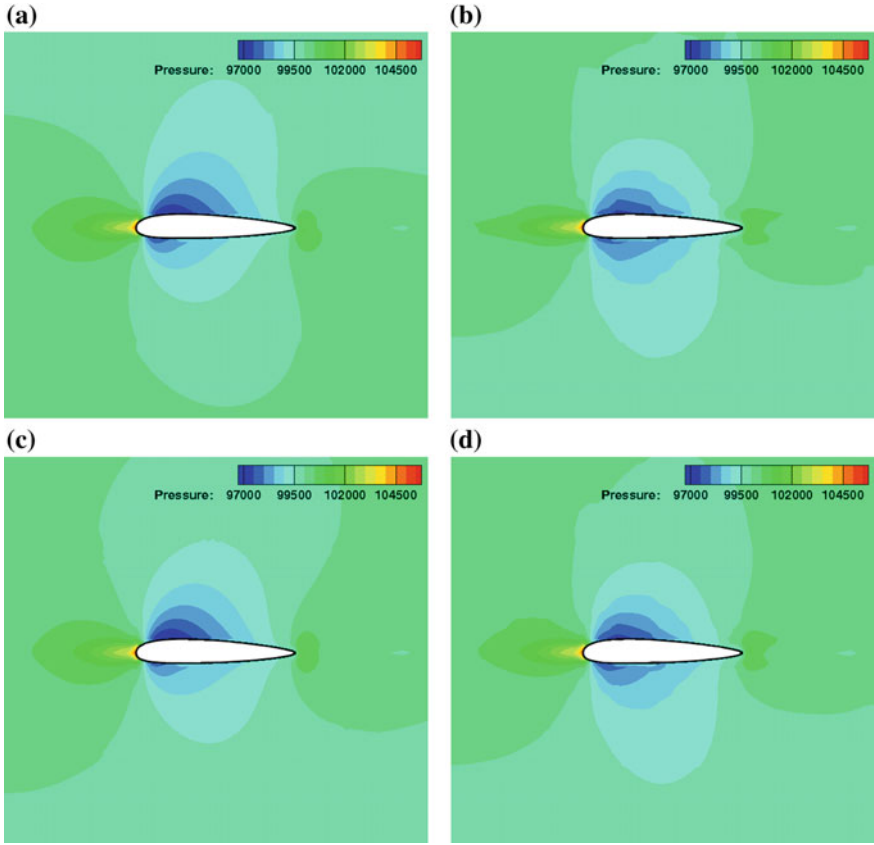


Fig. 4 Application 2: Instantaneous pressure fields for $a_\infty = -3^\circ$ (a) and $a_\infty = +3^\circ$ (c) are quite close each other with the corresponding fields (b) and (d) computed by the iPGD method

Having made both the primal and the adjoint flows available, the sensitivity derivatives of F w.r.t. b_q are computed. Figure 6 shows the effect on approximating the primal solution through the iPGD method on the accuracy of sensitivity derivatives. In the same figure, sensitivities computed by the posteriori PGD (i.e not incremental) compression of the primal unsteady solution (fully stored just for this purpose) are also shown. The extra deviation due to the incremental algorithm is much smaller than the total difference between the a posteriori PGD and the reference (from full storage) values of the derivatives, demonstrating the capabilities of the proposed incremental algorithm. Note that the only reason we additionally ran the adjoint based on the a posteriori PGD'ed primal solutions is for obtaining a good indication of the best accuracy we could ideally get by the iPGD.

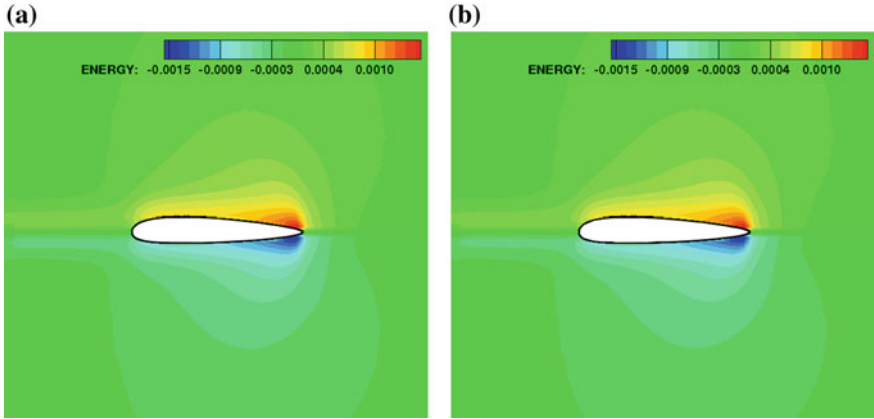
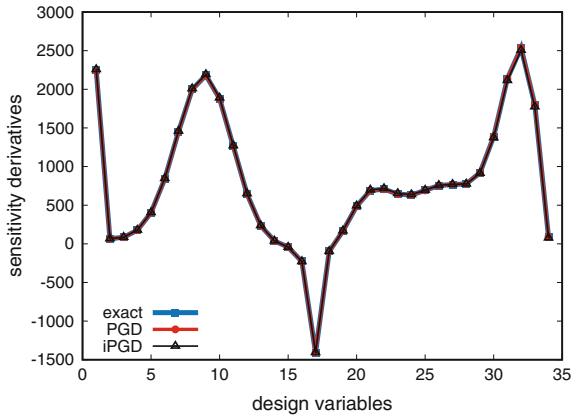


Fig. 5 Application 2: Instantaneous adjoint energy fields for $\alpha_\infty = -3^\circ$ based on the flow solution by the cut-cell method (a) and the iPGD-based approximation to the unsteady flow solution (b)

Fig. 6 Application 2: Comparison of sensitivity derivatives computed by the adjoint using a full storage, b the a posteriori PGD'ed primal solution and c the iPGD'ed one



Application 3: Pitching Airfoil

In the pitching airfoil case, since the mesh is changing in time, its data structure should have also been stored at each and every time-step over and above to the unsteady flow solution. However, the special structure of the Cartesian meshes allows minimum data storage, overcoming the need of compressing also the time-changing mesh. The airfoil is oscillating around the point at chord/4 with position angle that is a sinusoidal function with amplitude (3°) and period equal to $T_a = 0.015$ s, split into 20 time-steps. The average number of cells used at all time-steps is about 7000. The size of the RR mesh is 512×512 . The far-field Mach number is equal to 0.3.

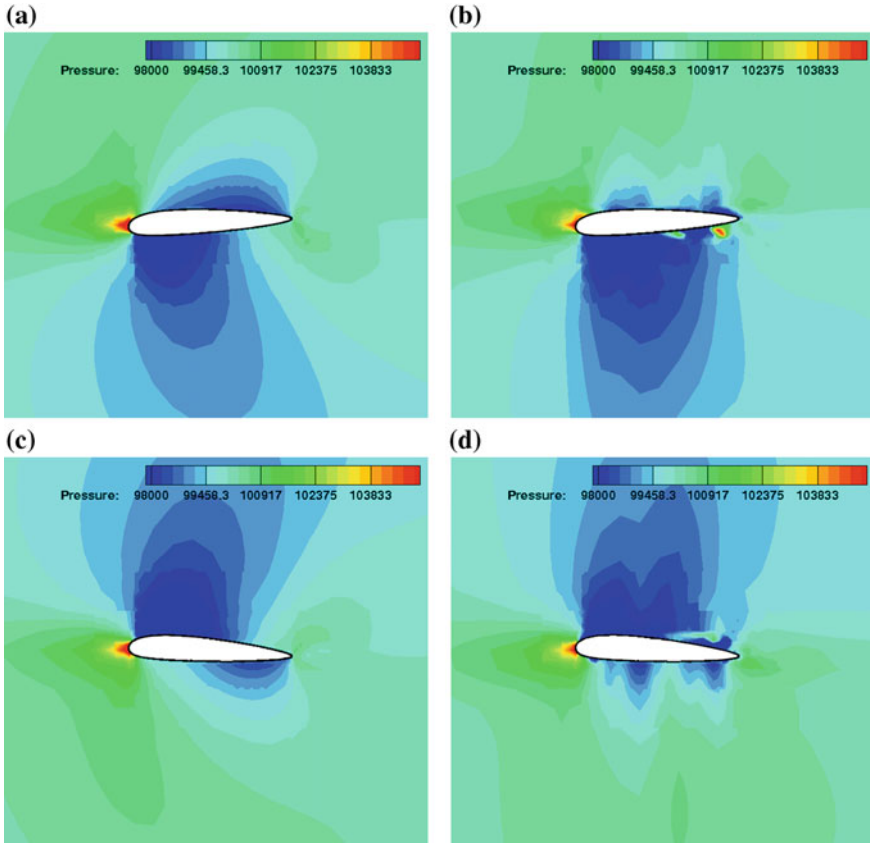


Fig. 7 Application 3: Instantaneous pressure fields at the lowermost (a) and the uppermost (c) positions of the airfoil motion are almost the same with the corresponding fields (b and d) computed by the iPGD method

The flow field is compressed via the iPGD algorithm using $M = 10$ modes. In Fig. 7 two fields are shown at the extreme instants of the period for the exact and the reconstructed fields.

The impact of the compressed primal fields on the adjoint solution was examined by solving the adjoint equations twice with full storage and the iPGD'ed primal data. Results of the adjoint solver are shown in Fig. 8.

For the two aforementioned cases, the sensitivity derivatives are computed, Fig. 9. Moreover, two extra curves for 20 and 30 modes are shown. As the number of modes increases, the primal and adjoint fields match each other much better and the error in the computed derivatives diminishes. Theoretically, by increasing the number of modes, the sensitivity derivatives should tend to the “exact ones” (the one from full storage). However, in practice they seem to stagnate since, in order to save computational cost we end up with a reasonably low number of modes.

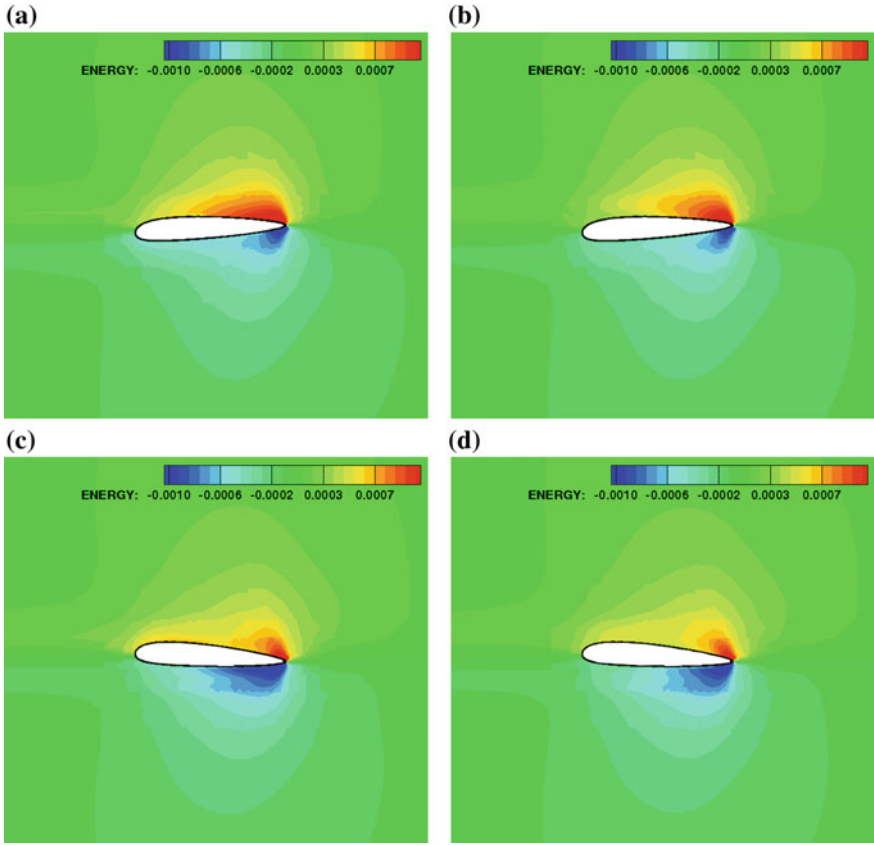
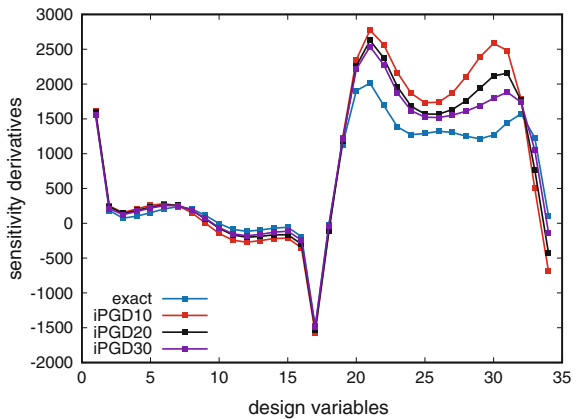


Fig. 8 Application 3: Instantaneous adjoint energy fields in the lowermost (a) and the uppermost (c) position of its motion are almost the same with the corresponding fields (b and d) computed by the iPGD

Fig. 9 Application 3: Comparison of the sensitivity derivatives computed using full storage and the iPGD’ed primal solution with $M = 10, 20, 30$. The fact that the sensitivity of the last design variable is approximated with the wrong sign is minor since this corresponds to the trailing edge which remains still



The saving in memory by the usage of the proposed iPGD algorithm is remarkable. The full storage of the unsteady flow field needs an average of $7000 \times 20 = 140000$ values to be saved in memory whereas, even with $M = 30$, this number drops to $30(512 + 512 + 20) = 31320$ with the iPGD. The reason for refraining to compare with checkpointing is exposed at the end of Section “[Application 1: Optimization Based on the Unsteady Heat Conduction Equation](#)”.

Conclusions

The use of the PGD within adjoint-based optimization, for time-varying problems was presented. The role of PGD is to approximate the time-series of the solution to the primal PDEs to support the integration of the adjoint equations, backwards in time. For the first time in the relevant literature, at least to the authors’ knowledge, an incremental PGD algorithm is proposed. Its distinguishing feature is that there is no need to store the time-series of the primal solution before processing them with the PGD. Instead, all modes are updated upon completion of a single step of the time-integration of the primal PDEs; more precisely, all spatial modes are updated to account for the new instantaneous primal field and a new element is appended to each one of the growing temporal modes. With the iPGD method, storage requirements are much lower compared to the full storage; no comparison with the checkpointing technique has been made since, the extra computation cost of partial recomputations of the primal solution of checkpointing should also be accounted for and compared with the extra cost of running the iPGD algorithm. We refrained from doing so since work on the acceleration of the iPGD method is in progress. The mathematical formulation of the new iPGD method is provided. Unsteady adjoint runs supported by the iPGD were demonstrated in unsteady heat and aerodynamic optimization problems and proved to offer a great economy in storage requirements. Even though all applications presented in this paper relied upon the continuous adjoint, the iPGD can also be used with discrete adjoint, as it is not related to the way the adjoint problem is formulated and solved. Another contribution of this work is the use of the iPGD with the cut-cell method, in which case the CFD meshes dynamically adapt to the shape boundaries. However, even in this case, it suffices to use a Reference/Regular mesh and appropriate interpolation schemes in order to be able to apply the iPGD as with standard meshes.

References

- Ammar A, Chinesta F, Cueto E, Doblar M (2012) Proper generalized decomposition of time-multiscale models. *Int J Numer Methods Eng* 90(5):569–596
- Balzano L, Wright S (2013) On grouse and incremental SVD. In: *Computational advances in multi-sensor adaptive processing (CAMSAP)*, IEEE 5th International Workshop. St. Martin, Dec, p 2013

- Chinesta F, Keunings R, Leygue A (2014) The proper generalized decomposition for advanced numerical simulations. A primer, Springer, Nantes
- Clarke D, Hassan H, Salas M (1986) Euler calculations for multielement airfoils using cartesian grids. *AIAA J* 24(3):353–358
- Griewank A, Walther A (2000) Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Trans Math Softw (TOMS)* 26(1):19–45
- Ladevèze P (2014) PGD in Linear and Nonlinear Computational Solid Mechanics. Springer, Vienna
- Papadimitriou D, Giannakoglou K (2007) A continuous adjoint method with objective function derivatives based on boundary integrals for inviscid and viscous flows. *Comput Fluids* 36(2):325–341
- Roe P (1981) Approximate Riemann solvers, parameter vectors, and difference schemes. *J Comput Phys* 43(2):357–372
- Samouchos K, Katsanoulis S, Giannakoglou KC (2016) Unsteady adjoint to the cut-cell method using mesh adaptation on GPUs. In: ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering, Crete island, Greece, June 5–10
- Vezyris C, Kavvadias I, Papoutsis-Kiachagias E, Giannakoglou K (2016) Unsteady continuous adjoint method using POD for jet-based flow control. In: ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Sciences and Engineering, Crete island, Greece, June 5–10

A Two-Step Mesh Adaptation Tool Based on RBF with Application to Turbomachinery Optimization Loops



Flavio Gagliardi, Konstantinos T. Tsiakas and Kyriakos Giannakoglou

Abstract Adapting an unstructured CFD mesh to the modified geometry, in accordance with the updated value-set of design parameters at the end of each cycle, is a must in CFD-based shape optimization loops. Mesh adaptation is a nice alternative to remeshing procedures which might become expensive and, also, hinder the initialization of new simulations from previous results. Mesh morphing, based on Radial Basis Functions (RBF) network, has been widely used in the past to smoothly propagate boundary nodal displacements into the volume mesh while preserving its validity and quality. To precisely capture even small design changes, all surface mesh nodes must be used as interpolation nodes which, in case of large meshes for real-world application, leads to excessive computational cost and memory requirements. This paper introduces a cost reduction strategy for mesh adaptation, by proposing a new two-step RBF interpolation employing the Sparse Approximate Inverse (SPAI) preconditioner and the Fast Multipole Method (FMM). The software is demonstrated in the aerodynamic shape optimization of a turbomachinery row. The purpose of this paper is not to solve the optimization problem itself; emphasis is laid on the way the proposed method may handle large displacements and, for this reason, Evolutionary Algorithms (EA) which allow great variations in the values of the design variables were first used. Adjoint-based optimization follows; its role is to perform the refinement of the best solution obtained by the EA-based search.

F. Gagliardi (✉) · K. T. Tsiakas · K. Giannakoglou
Parallel CFD & Optimization Unit, School of Mechanical Engineering,
National Technical University of Athens (NTUA), Athens, Greece
e-mail: fl.gagliardi@gmail.com

K. T. Tsiakas
e-mail: tsiakost@gmail.com

K. Giannakoglou
e-mail: kgianna@central.ntua.gr

RBF-Based Mesh Displacement: Introduction & Literature Survey

To perform an automated CFD-based aerodynamic shape optimization, a flow solver, a shape parameterization method, an optimization technique and a procedure to adapt or regenerate the CFD mesh for each new candidate solution must be available.

The method presented in this paper focuses onto the problem of CFD mesh adaptation and is demonstrated in the optimization of a compressor stator. In specific, an RBF-based mesh adaptation technique, able to smoothly propagate the displacements of all surface mesh nodes to the interior of the domain has been devised and programmed. This requirement springs from the need of adapting an existing mesh to an updated CAD representation of the shape to be optimized. This can be used in optimizations which employ a CAD system to build the geometry, with the CAD parameters as design variables. In this paper, an in-house parameterization/design software for turbomachinery bladings (Tsiakas et al. 2016) is used to generate a NURBS-based representation of the geometry.

Obtaining a new surface mesh conforming with the changed boundaries is the starting point for deforming the volume mesh. This paper, however, focuses only on the adaptation of the volume mesh given the displacements of the surface mesh nodes which are obtained by inverting and displacing nodes in the NURBS parametric space, taking special care for trimmed surfaces (Tsiakas et al. 2016).

RBF-based interpolation methods are robust but may become computationally expensive, especially for large meshes. In Section “[Step 1: Predictor](#)”, it is shown that, by using a data-reduction algorithm, fewer nodes are used to approximate the new shape, reducing dramatically the computational cost and memory requirements compared to the standard formulation. However, this is expected to deteriorate the geometrical precision of boundaries. In Section “[Step 2: Corrector](#)”, a strategy to recover the deviation of the surface mesh with respect to the prescribed shape, caused by the previously made approximation, is proposed.

The theory of RBF networks, Buhmann (2009), is briefly summarized below. RBF networks can *interpolate* discrete data in the n -dimensional space. In mesh displacement, quantities to be interpolated are the known displacements defined at *source nodes* or *RBF centres*. An RBF deformation function $\mathbf{d} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a linear combination of radially symmetric kernels $\phi_s(\mathbf{y}) = \phi(\|\mathbf{x}_s - \mathbf{y}\|)$ ¹ centered at the N *source nodes* $\mathbf{x}_s \in \mathbb{R}^3$ and weighted by $\mathbf{w}_s \in \mathbb{R}^3$:

$$\mathbf{d}(\mathbf{y}) = \sum_{s=1}^N \mathbf{w}_s \phi_s(\mathbf{y}) \quad (1)$$

where \mathbf{y} is the *target node* position vector. All M (boundary and internal) mesh nodes \mathbf{y}_t , $t \in [1, \dots, M]$, for which Eq. 1 provides their displacements $\mathbf{d}(\mathbf{y}_t)$ are

¹ $\|\dots\|$ is the Euclidean norm.

considered as *target nodes*. The N boundary mesh nodes with known displacements $\boldsymbol{\delta}_s \in \mathbb{R}^3$, $s \in [1, \dots, N]$ are used as *source nodes* \mathbf{x}_s .

Weights \mathbf{w}_s are computed so as to exactly reproduce the imposed displacements $\boldsymbol{\delta}_s$ at source nodes; this requires the numerical solution of a linear system with an $N \times N$ symmetric positive-definite² coefficient matrix \mathbf{A} , namely:

$$\begin{bmatrix} \phi_1(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \cdots & \phi_N(\mathbf{x}_N) \end{bmatrix} \begin{pmatrix} \mathbf{w}_1^T \\ \vdots \\ \mathbf{w}_N^T \end{pmatrix} = \begin{pmatrix} \boldsymbol{\delta}_1^T \\ \vdots \\ \boldsymbol{\delta}_N^T \end{pmatrix} \quad (2)$$

The computation of weights \mathbf{w}_s , by solving Eq. 2, is the most computationally expensive task. It shows poor scalability if implemented naively, due to both the complexity of linear solvers and its stiffness. After solving Eq. 2, the displacements $\mathbf{d}(\mathbf{y}_i)$ for all mesh nodes \mathbf{y}_i are computed by Eq. 1 at the cost of $M \times N$ RBF kernel evaluations. The behaviour of the RBF interpolation is highly influenced by the chosen kernel ϕ (de Boer et al. 2007).

Mesh adaptation based on RBF interpolation has been standing out in the literature for their wide range of application. Selim and Koomullil (2016) discuss advantages and disadvantages of the most widely used techniques, such as linear and torsional springs, linear elasticity and several interpolation based methods. Based on their work, mesh deformation based on RBF interpolation is one of the most promising approach in terms of robustness and morphed mesh quality, on condition that its high computational cost and bad scalability can be mitigated by methods such as greedy data reduction algorithms.

Greedy algorithms (Hon et al. 2003) start from a coarse approximation to the deformation and iteratively refine it until the desired accuracy be reached. They use a subset of the surface mesh nodes to describe the new shape, leaving the rest of the nodes for error checking. These methods are more efficient than standard RBF interpolation but they cannot reproduce exactly all surface deformations. The iterative procedure, required to guarantee the error drop to a prescribed tolerance is, for tight surface tolerances, time consuming. Some authors also suggested to apply a correction step such as an explicit interpolation (Rendall and Allen 2010) or Delaunay graph mapping (Wang et al. 2015) after the approximation step, but locally supported RBF interpolation appears to be a better choice from the quality and robustness point of view (Gillebaart et al. 2016). Other methods aiming at reducing the RBF-based interpolation cost can be found in the literature. RBF multilevel techniques involve successive levels of nested RBF interpolations where, at each level, the solution from the previous coarser level is interpolated (Narcowich et al. 1999; Lazzaro and Montefusco 2002; Floater and Iske 1996). In this paper, the interpolation is practically carried out on two levels, with the advantage of being able

²Under certain conditions explained in Buhmann (2009).

to use other cost reduction methods, which have a dominant setup time and would be impractical to use in many levels. In Kedward et al. (2017), a multiscale RBF interpolation which uses multiple support radii to capture deformations at different scales is presented. The interpolation matrix is built starting from a coarse subset of source nodes. The algorithm proceeds by iteratively adding the remaining source nodes using a support radius such that the newly added nodes do not affect the previous, ending up with an easier to solve linear system. In such a method, the necessary preprocessing phase cost is dominant, and it is suggested to be performed once before all mesh adaptations.

The Proposed Two-Step RBF Mesh Displacement Strategy

The proposed method works by hierarchically using an approximate predictor step followed by a correction one. Both rely on RBF networks. The two steps are briefly described as follows:

- In the first step (*predictor*), all mesh nodes (surface, interior) become interpolation targets and a new coarsened set of source nodes is generated by a non-iterative data reduction method. This method is *adaptive*, in the sense that the data reduction is performed by taking into account the displacement field to be interpolated instead of just the spatial distributions of mesh nodes. The entire mesh is displaced; however, boundary mesh nodes do not precisely respect the known boundary displacements since a reduced number of sources is used.
- The second step (*corrector*), corrects the position of the surface mesh nodes, through local deformations.

The first step generates a “small” but dense coefficient matrix (its rank might be by orders of magnitude lower than the number of surface mesh nodes) while the second generates a “big” (rank equal to the number of surface mesh nodes), though very sparse, matrix.

Step 1: Predictor

The predictor is an RBF approximation tool based on data reduction according to which the source nodes are coarsened by clustering. Cost reduction does not rely only on the reduced problem size but, also, on the implementation of methods such as the Sparse Approximate Inverse (SPAI) preconditioner (Kallischko 2007) and the Fast Multipole Method (FMM) (Fong and Darve 2009). The predictor is divided into three sub-steps: data reduction, training and application, which are described below.

Data Reduction

The objective of the data reduction phase is to find a reduced set of RBF centres \mathbf{x}_r , $r \in [1, \dots, N_R]$ which is representative of the displacement field of the surface mesh nodes \mathbf{x}_s , $s \in [1, \dots, N_S \gg N_R]$. For this purpose, an *adaptive octree* data structure is employed, which recursively splits the Cartesian space. By taking into account the surface mesh nodes \mathbf{x}_s density and the spatial gradient of the displacements $\nabla \delta_s$ the collocation of more RBF centres in areas of rapid variation of the imposed displacements is ensured.

The centres of leaf (childless) octree boxes \mathbf{x}_r are used as RBF centres in the predictor training step. The displacement δ_r of each RBF centre \mathbf{x}_r is the averaged displacement of the source nodes \mathbf{x}_s contained in the corresponding leaf box of the octree. Such a method does not allow the error to be estimated a priori or reduced iteratively, but it quickly generates the reduced point clouds to approximate the displacement field. This approach is preferred since any error in the reproduction of the imposed displacements will be resolved in the following corrector step. Figure 1 shows an example of selected RBF centres and the corresponding CFD surface mesh.

RBF Network Training

The training process runs on the previously generated RBF centres \mathbf{x}_r . A global support RBF kernel is chosen taking into account different characteristics, such as mesh quality preservation (Rendall and Allen 2010), flop count, condition number of the generated linear system and smoothing effect. In this step, the following kernel is used

$$\phi(r) = \frac{1}{\frac{r}{\sigma} + 1} \quad (3)$$

where σ is the shape parameter regulating the width of the kernel and r the euclidean distance between two nodes. The linear system of Eq. 2, assembled with \mathbf{x}_r as RBF centres, is solved by an iterative method.

A *Sparse Approximate Inverse* (SPAI) (Kallischko 2007) preconditioner is implemented to speed-up the convergence of the iterative solver. This preconditioner is, in general, not symmetric, and a solver for non-symmetric matrices i.e., Bi-Conjugate-Gradient-Stabilized (BiCGStab), has to be used. The SPAI preconditioner \mathbf{M} is an approximate inverse of an approximation to \mathbf{A} (Eq. 2). The method is based on the minimization of the Frobenius norm

$$\min_{\mathbf{M}} \|\mathbf{SM} - \mathbf{I}\|_F^2 \quad (4)$$

where \mathbf{I} is the identity matrix and \mathbf{S} is a sparse matrix formed by the largest entries of \mathbf{A} .

The sparsity pattern of \mathbf{S} is defined a priori through a sparsification strategy based on geometric considerations (Carpentieri 2009), avoiding thresholding strategies: for each RBF centre, all other centres in the neighborhood, from which the biggest entries of \mathbf{A} arise, are selected as entries of \mathbf{S} . Thanks to the decaying RBF kernels, the largest entries of \mathbf{A} are arranged in bands and the largest entries of \mathbf{A}^{-1} are expected to be at the same location with the largest entries of \mathbf{A} (Demko et al. 1984), so that, for \mathbf{M} , the same or a similar sparsity pattern to \mathbf{S} can be employed. Figure 2 shows the pattern of the large entries of an RBF training matrix \mathbf{A} and its inverse (\mathbf{A}^{-1}), for the CFD mesh of the turbomachinery case.

In Eq. 4, the computation of \mathbf{M} is based on a property of the Frobenius norm that allows to split it into a sum of Euclidean norms

$$\|\mathbf{SM} - \mathbf{I}\|_F^2 = \sum_i^{N_R} \|\mathbf{S}\mathbf{m}_i - \mathbf{e}_i\|_2^2 \quad (5)$$

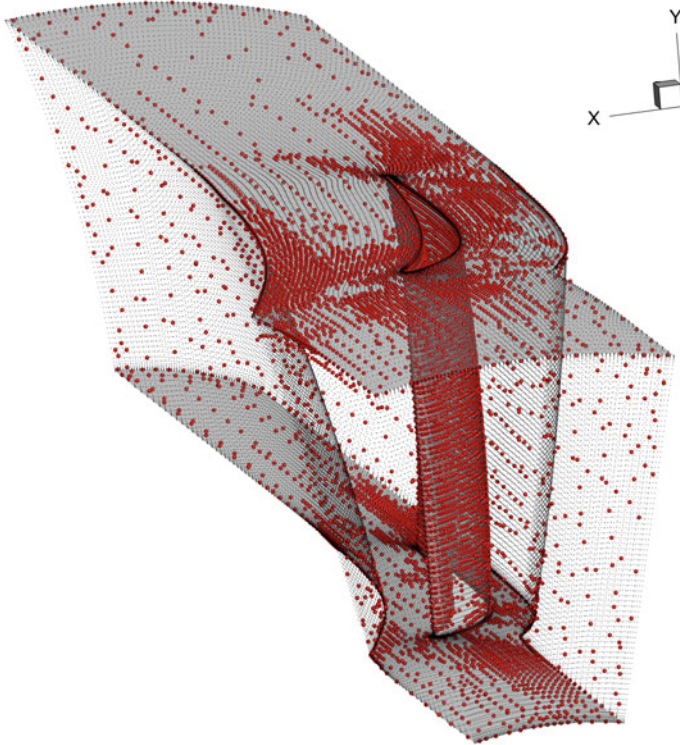


Fig. 1 RBF centres \mathbf{x}_r (red spheres) generated by the data reduction algorithm in the predictor step (for a certain displacement of the CFD mesh on the compressor blade). Original mesh surface nodes \mathbf{x}_s are displayed as black dots. More RBF centres lie in the area of high spatial gradient of the displacements $\nabla\delta_s$. Generally, the RBF centres \mathbf{x}_r do not lie on the mesh surface

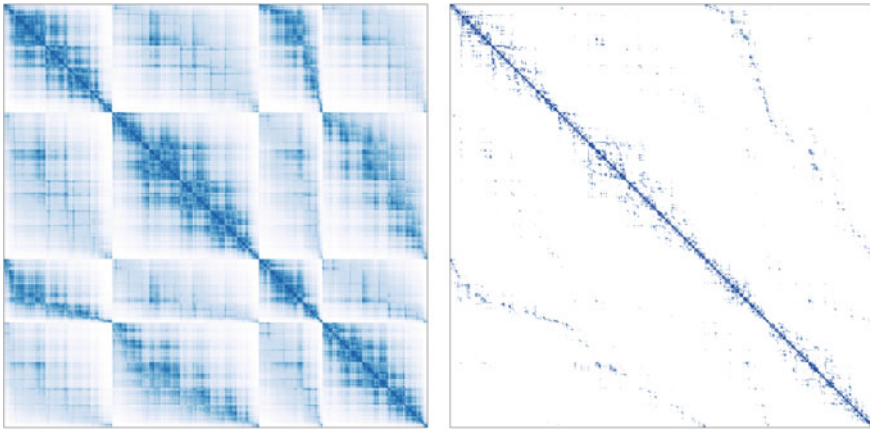


Fig. 2 Pattern of the large entries in a predictor RBF training matrix \mathbf{A} (left) and its inverse \mathbf{A}^{-1} (right). Large to small entries are shown from blue to white. The matrix rank is $\sim 10^4$ and is originated from the turbomachinery case shown in Fig. 1

where \mathbf{m}_i and \mathbf{e}_i are the i th columns of \mathbf{M} and \mathbf{I} . Each summand in Eq. 5 constitutes a linear system, which is solved separately from each other with Cholesky decompositions. The rank of each linear system is noticeably reduced thanks to the sparsity of \mathbf{m}_i . The number of decompositions needed is also significantly reduced using geometric considerations: in fact, all RBF centres in the same neighborhood, identified by an integer lattice³, will lead to the same reduced linear system which is decomposed just once (Carpentieri 2009). This procedure overestimates the fixed radius neighbors which leads to bigger linear systems. However, this is compensated by the reduced number of linear systems to be solved, reduced complexity in the neighbors search and higher quality of the preconditioner due to the greater number of entries.

Figure 3 shows the time required for the solver to converge including the setup time for various preconditioners.

RBF Network Application

After having solved Eq. 2 for the weights \mathbf{w}_r , the interpolated displacement field results from Eq. 1 applied at each mesh node. For large meshes the application time can noticeably be reduced using the Fast Multipole Method (FMM) (Fong and Darve 2009). FMM is an algorithm for approximating sums such as those appearing in Eq. 1, with reduced complexity and controllable error. Briefly, the FMM exploits the decay of the RBF kernel by computing interactions between mesh nodes on different

³An integer lattice is tessellation of the \mathbb{R}^3 euclidean space in bricks. It is equivalent to a level of an octree.

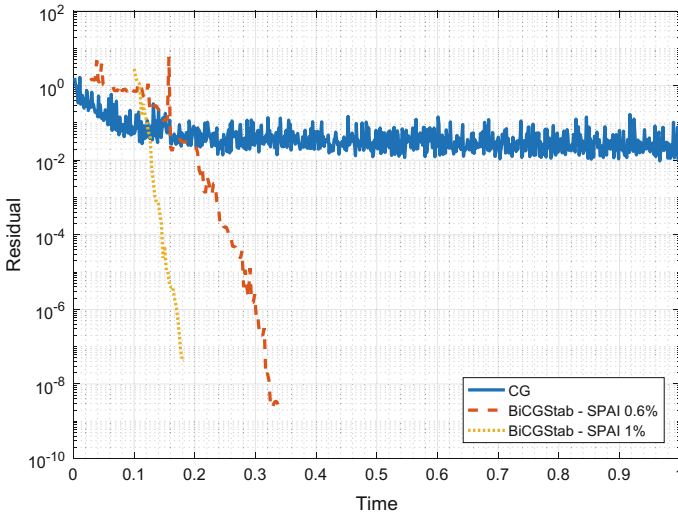


Fig. 3 History of the residual of a full linear system (rank $\sim 4 \times 10^4$) for various combinations of iterative solvers and SPAI preconditioners plotted as the function of normalized time. In the sake of fairness, the setup time for the preconditioners, which appears as a delay before the solvers take over, is considered too. Percentages in the legend refer to the density (which is equal to 1 minus the sparsity of the matrix) of the preconditioners. The non-preconditioned CG solver does not have any setup time but the convergence rate is badly affected by the system ill-conditioning. Two different SPAI preconditioners were used, with different densities to show that a correlation exists between density and quality but, of course, a denser preconditioner requires more time to be built

levels of accuracy depending on their geometric distances. This is achieved by low-rank approximations to the displacement field in conjunction with a hierarchical decomposition of the Cartesian space. The quality of the low-rank approximations determines the error made by the method.

There is a trade-off between complexity and approximation error and whether this approach becomes advantageous or not depends on the mesh size but, also, the minimum mesh nodal distance, which determines the maximum allowed multiplication error. In fact, the risk is to introduce a great error (due to the prescribed tolerance) in the interpolated displacements which could yield critical mesh elements quality. Figure 4 shows the time required to perform RBF network applications for increasing mesh sizes with and without the FMM. The FMM-based RBF network application is cheaper for big meshes.

The implementation relies on the black-box FMM (bbFMM) (Fong and Darve 2009). It is “black-box” in the sense that the functional form of the low-rank approximation is independent of the RBF kernel used since this is based on polynomial interpolation on Chebyshev nodes.

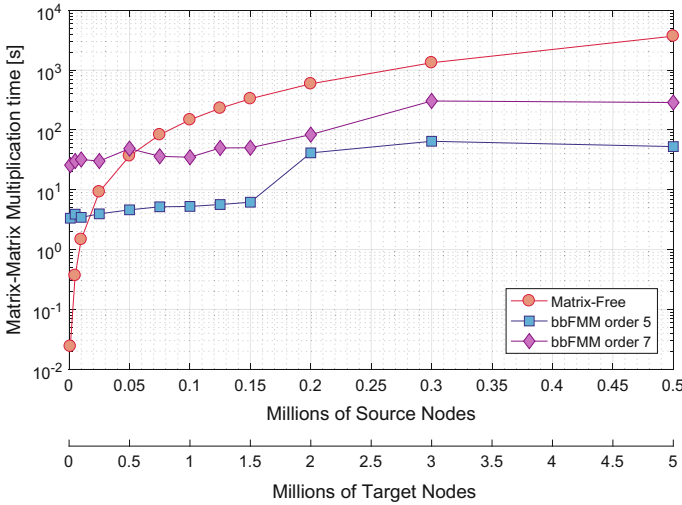


Fig. 4 Wall clock time for the evaluation of Eq. 1 by varying the number of source and target nodes (always in the ratio 1:10) using the bbFMM method for the RBF kernel of Eq. 3. The FMM-based multiplications include the FMM setup time. Two interpolation orders, 5 and 7, are shown for the bbFMM, introducing a maximum approximation error (infinity norm) smaller than 1×10^{-5} and 1×10^{-7} , respectively. Measurements were performed on a computational node with two 6-core Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10 GHz processors

Step 2: Corrector

The corrector step is based on a local RBF interpolation method. Locality is ensured by the kernel formulation, which vanishes when the distances r of two mesh nodes is higher than the so-called *local support radius* r_s . The locally supported RBF kernel used in this step is the Wendland C0 function (Wendland 1995):

$$\phi(r) = \begin{cases} \left(1 - \frac{r}{r_s}\right)^2 & \text{if } r < r_s \\ 0 & \text{if } r \geq r_s \end{cases} \quad (6)$$

A tradeoff between the smooth propagation of the deformations in the volume mesh, computation time and memory requirements, depending on the choice of the support radius, is expected. Lower support radius leads to better conditioned and sparser training matrix (see Eq. 2) whereas deformation is dissipated in a smaller portion of the interior mesh.

In the corrector, the RBF centres coincide with the surface mesh nodes with prescribed displacements. Since the predictor has already displaced the surface mesh nodes close to their target positions, the remaining surface displacements are relatively small and a small local support radius can be chosen.

The concept of *fixed-radius neighbors search* is used to reduce the matrix filling time in Eq. 2. An integer lattice, scaled so that the distance between lattice points is r_s , is built to map nearby to each other mesh nodes. The lattice is, then, used to compute non-zero interactions $\phi(r)$ between close nodes instead of all pairwise interactions.

After solving the sparse linear system, with the help of the SPAI preconditioner, the displacement field is obtained by evaluating Eq. 1 at all mesh nodes. By searching in a fixed-radius area, the computation of null kernel values of Eq. 1 is avoided.

Compressor Stator Blade Optimization Results

The two-step mesh adaptation tool is tested by performing an EA-based, followed by an adjoint-based, optimization of the blade shape of the TU Berlin TurboLab low-speed compressor stator (TUB 2016). Inlet conditions are provided in the form of radial profiles, corresponding to 39.7° average inlet flow angle w.r.t. the axial direction, 104 kPa average inlet total pressure and 301 K average inlet total temperature. The outlet static pressure is adjusted to impose 9.5 kg/s full-annulus mass flow rate.

The objective is to minimize the mass-averaged deviation of the exit flow from the axial direction, defined as

$$\alpha = \left(\frac{\int_{S_o} \left(\cos^{-1} \left(\frac{V_a}{|\mathbf{V}|} \right) \right)^2 \rho V_a \, dS}{\int_{S_o} \rho V_a \, dS} \right)^{\frac{1}{2}} \quad (7)$$

where V_a is the axial component of the velocity, $|\mathbf{V}|$ the velocity magnitude, ρ the density and S_o and S_I the stator outlet and inlet sections.

The in-house GPU enabled RANS solver for compressible flows (Asouti et al. 2011), employing the Spalart-Allmaras turbulence model, and its adjoint were used. The blade is parameterized with the in-house turbomachinery row CAD software (Tsiakas et al. 2016), with 133 design variables. The mesh is block-structured with $\sim 2.2 \times 10^6$ nodes.

In Section “[Mesh Adaptation to the Displaced Boundaries](#)”, the performance of the mesh adaptation model is discussed. Optimization results follow in Section “[Aerodynamic Shape Optimization Results](#)”.

Mesh Adaptation to the Displaced Boundaries

The mesh adaptation tool is tested by displacing the initial compressor stator mesh to the improved design generated by the hybrid optimization. Figure 5 shows the

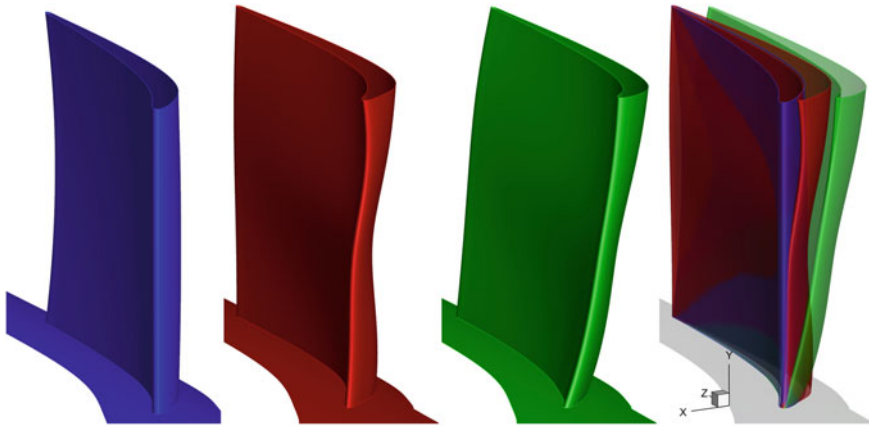


Fig. 5 EA and adjoint-based optimization of a low-subsonic compressor stator blade: reference geometry (1st), geometry generated by the EA-based optimization (2nd) which was used to explore the design space before switching to the adjoint-based optimization (3rd). On the right (4th), the shapes of the three blades are superimposed

Table 1 Quality metrics for the reference and adapted block structured volume mesh ($\sim 2.2 \times 10^6$ nodes) in the optimal geometry resulted from the adjoint-based optimization and shown in Fig. 5. The sign of the Jacobian is used to check the validity of the mesh. Larger orthogonality metric values and lower normal skewness values are desirable to avoid deteriorating the CFD solution accuracy and robustness. Max. $y^+ < 1$ of the first nodes of the wall is required to guarantee that the mesh near the wall is adequate for CFD simulations

	Reference	Adapted
Min. Jacobian	>0	>0
Min. Orthogonality	0.144	0.115
Avg. Orthogonality	0.79	0.76
Max. Normal Skewness	0.856	0.885
Avg. Normal Skewness	0.21	0.24
Max. y^+	0.50	0.51

reference mesh and those resulting from the optimization runs. Table 1 reports quality metrics for the reference and adapted meshes (resulting from the adjoint-based optimization).

Table 2 tabulates metrics showing the deviation (distance) of the reference and displaced surface meshes at each step of the mesh adaptation procedure. The first step reduces significantly the deviation but the resulting surface mesh does not perfectly fit to the new geometry. This is corrected during the second step.

Figure 6 shows an analysis conducted in order to investigate the time and memory requirements to perform a mesh displacement for growing mesh sizes. The predictor

Table 2 Deviation metrics for the reference and displaced CFD surface meshes ($\sim 1.20 \times 10^5$ nodes), resulting from the adjoint-based optimization, Fig. 5. The first column lists the values for the surface mesh deviation prior to mesh adaptation. Columns labeled “1st and 2nd Step” give the surface mesh deviation upon completion of the corresponding steps

	Initial	1st Step	2nd Step
Infinity Norm	3.27×10^{-2}	5.30×10^{-4}	4.85×10^{-14}
Euclidean Norm	3.83	1.83×10^{-2}	1.51×10^{-12}
Avg. Deviation	4.87×10^{-3}	2.31×10^{-5}	2.57×10^{-15}

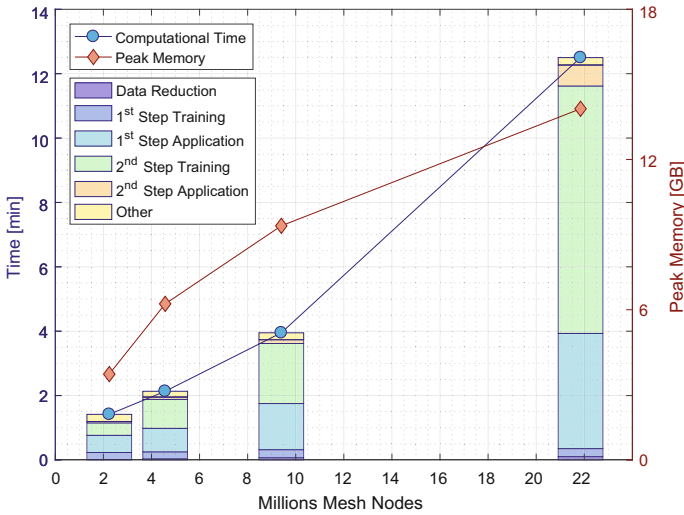


Fig. 6 Computational time and max. RAM memory requirement for mesh displacements, by varying the mesh size. The computational time is broken down in the five main steps in the bar chart: Clustering (Section “Data Reduction”), predictor training (Section “RBF Network Training”) and application (Section “RBF Network Application”) as well as corrector training and application (Section “Step 2: Corrector”). The bar chart and computational time refer to the left vertical axis, while the peak memory curve points to the right one. Measurements are performed on a computational node with two 6-core Intel(R) Xeon(R) CPU E5-2620 v2 @ 2.10 GHz processors

application and corrector training are the most expensive ones. The former scales linearly with mesh size, thanks to the Fast Multipole Method. The latter scales super-linearly due to the BiCGStab computational complexity and increased matrix size and setup time of the SPAI preconditioner. The predictor training time is almost constant since the imposed surface mesh displacements are similar for all mesh sizes. The corrector application phase also scales super-linearly due to the increased number of RBF kernel evaluations, but its contribution to the total required time remains minimal thanks to the computation strategy which is based on the integer lattice.

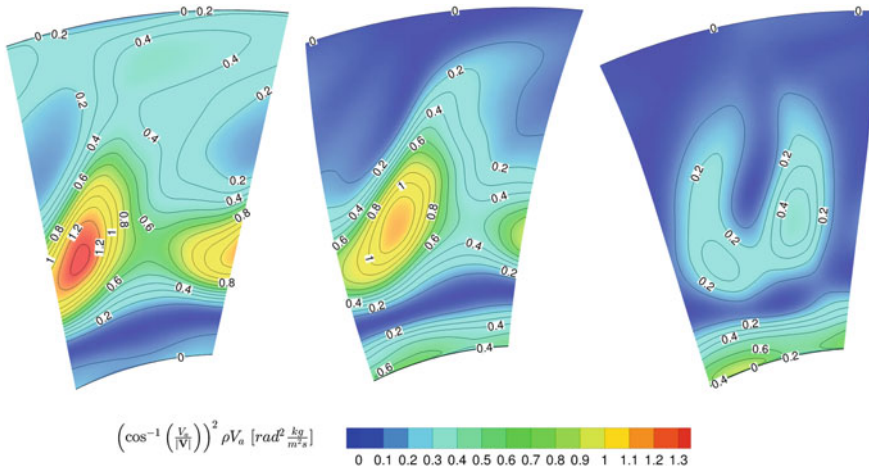


Fig. 7 $\left(\cos^{-1}\left(\frac{V_a}{|V|}\right)\right)^2 \rho V_a$ field (see Eq. 7) at the stator outlet for the reference (left), best solution from the EA (centre) and best solution from the adjoint-based optimization (right)

Aerodynamic Shape Optimization Results

The EA-based optimization was performed using the s/w EASY (Evolutionary Algorithm System) (2008) developed by the NTUA group. Only the 30 most important design variables of the geometry, reparameterized by the in-house turbomachinery row parameterization software, were used. The adjoint-based optimization relied upon Sequential Quadratic Programming (SQP). The turbomachinery row parameterization software was differentiated and coupled with the in-house GPU-enabled continuous adjoint solver, for computing the gradients (Tsiakas et al. 2016).

The initial design shows a deviation of the exit flow from the axial direction of 5.52° . The EA-based design space exploration was able to reduce it to 3.98° at the cost of 150 CFD simulations. The adjoint-based optimization resulted to an even better solution with α equal to 3.16° , at the cost of 40 equivalent CFD simulations. The deviation of the exit flow from the axial direction for the reference and optimized blades are shown in Fig. 7.

Conclusions

An efficient mesh adaptation method, based on RBF, that uses all surface mesh nodes to ensure an exact surface representation was presented. This is achieved by using a correction step, to move all surface mesh nodes to their exact positions, following an approximate step (predictor), taking care of the largest part of the displacements.

These steps are furthermore accelerated using the SPAI preconditioner based on geometric considerations and the Fast Multipole Method.

The reliability of the two-step strategy in cases with relatively large displacements and the significant scalability, in terms of computational time and memory requirements for large meshes, have been shown.

During the hybrid optimization of a low-subsonic compressor stator, the proposed mesh adaptation method was successfully used, reducing the computational resources (compared to a re-meshing strategy) required to improve the row design and demonstrating its flexibility in handling large and small displacements. The results show that the proposed two-step mesh adaptation model has high efficiency and its cost scales almost linearly with the mesh size, preserving mesh quality consistently even in large design variations.

Acknowledgements The authors thank Dr. Varvara G. Asouti and Dimitrios H. Kapsoulis for their assistance during the EA-based optimization with the s/w EASY (Evolutionary Algorithm System) (2008) and Dr. S. Xenofon Trompoukis for his assistance with the in-house flow and adjoint solvers (Asouti et al. 2011).

This research was funded from the People Programme (ITN Marie Curie Actions) of the European Union's H2020 Framework Programme (MSCA-ITN-2014-ETN) under REA Grant Agreement no. 642959 (IODA project). The first author is an IODA Early Stage Researcher.

References

- AboutFlow Project Website: TU Berlin TurboLab Stator Case (2016) <http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3/>
- Asouti VG, Trompoukis XS, Kampolis IC, Giannakoglou KC (2011) Unsteady CFD computations using vertex-centered finite volumes for unstructured grids on graphics processing units. *Int J Numer Methods Fluids* 67(2):232–246
- Buhmann M (2009) *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, Cambridge Monographs on Applied and Computational Mathematics
- Carpentieri B (2009) Algebraic preconditioners for the fast multipole method in electromagnetic scattering analysis from large structures: trends and problems. *Electron J Bound Elem* 7(1):
- de Boer A, van der Schoot M, Bijl H (2007) Mesh deformation based on Radial Basis Function interpolation. *Comput Struct* 85(1114):784–795
- Demko S, Moss WF, Smith PW (1984) Decay rates for inverses of band matrices. *Math Comput* 43(168):491–499
- Floater MS, Iske A (1996) Multistep scattered data interpolation using compactly supported radial basis functions. *J Comput Appl Math* 73(1):65–78
- Fong W, Darve E (2009) The black-box fast multipole method. *J Comput Phys* 228(23):8712–8725
- Gillebaart T, Blom D, van Zuijlen A, Bijl H (2016) Adaptive radial basis function mesh deformation using data reduction. *J Comput Phys* 321:997–1025
- Hon Y, Schaback R, Zhou X (2003) An adaptive greedy algorithm for solving large RBF collocation problems. *Numer Algorithms* 32(1):13–25
- Kallischko A (2007) *Modified sparse approximate inverses (MSPA) for parallel preconditioning*. PhD thesis, Technische Universität München, Germany
- Kedward L, Allen CB, Rendall T (2017) Efficient and exact mesh deformation using multiscale RBF interpolation. *J Comput Phys* 345:732–751

- Lazzaro D, Montefusco LB (2002) Radial basis functions for the multivariate interpolation of large scattered data sets. *J Comput Appl Math* 140(1):521–536
- Narcowich FJ, Schaback R, Ward JD (1999) Multilevel interpolation and approximation. *Appl Comput Harmonic Anal* 7(3):243–261
- Rendall TCS, Allen CB (2010) Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors. *Int J Numer Methods Eng* 81(1):89–105
- Selim M, Koomullil R (2016) Mesh deformation approaches-a survey. *J Phys Math* 7(181):2090–2092
- The EASY (Evolutionary Algorithms SYstem) software (2008) <http://velos0.lt.mech.ntua.gr/EASY>
- Tsiakas KT, Gagliardi F, Trompoukis XS, Giannakoglou KC (2016) Shape optimization of turbomachinery rows using a parametric blade modeller and the continuous adjoint method running on GPUs. In: 7th ECCOMAS Conference Proceedings, Crete Island, Greece
- Wang Y, Qin N, Zhao N (2015) Delaunay graph and radial basis function for fast quality mesh deformation. *J Comput Phys* 294:149–172
- Wendland H (1995) Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv Comput Math* 4(1):389–396

Adjoint-Based Aerodynamic Optimisation of Wing Shape Using Non-uniform Rational B-Splines



Xingchen Zhang, Rejish Jesudasan and Jens-Dominik Müller

Abstract Numerical shape optimisation with adjoint CFD is applied using the NURBS-based parametrisation method with continuity constraints (NSPCC) for aerodynamically optimising three dimensional surfaces. The ONERA M6 wing is re-parametrised with NURBS surfaces including weight adjustments to represent the three dimensional wing accurately, resulting in fewer control points and smoother variation of curvature. The NSPCC CAD kernel is coupled with the in-house flow and adjoint solver STAMPS and a gradient-based optimiser to minimise the drag of the ONERA M6 wing in transonic Euler flow conditions. Optimisation results are presented for the B-Spline and NURBS parametrisations.

Introduction

Numerical shape optimisation has attracted widespread interest in recent years. This is mainly driven by engineering design facing ever-increasing requirements in performance, environmental impact and life-time cost. To satisfy these needs, large design spaces with many degrees of freedom need to be explored systematically, which can only be achieved through numerical optimisation.

Gradient-free optimisation methods such as Genetic Algorithms (GA) or Evolutionary Algorithms (EA) are very established, especially for linear structural optimisation where the computational cost of an evaluation is low. However, these methods

X. Zhang (✉) · R. Jesudasan · J.-D. Müller
School of Engineering and Materials Science,
Queen Mary University of London, Mile End Road, London E1 4NS, UK
e-mail: xingchen.zhang@qmul.ac.uk

R. Jesudasan
e-mail: r.jesudasan@qmul.ac.uk

J.-D. Müller
e-mail: j.muller@qmul.ac.uk

require a large number of function evaluations when going beyond 50–100 design variables, and this cost becomes prohibitive when used with expensive CFD models.

The convergence of gradient-based methods on the other hand suffers much less from large design spaces and they have been adopted as the method of choice for shape and topology optimisation with CFD. This replaces the hurdle of computational cost with the challenge to compute the gradients for all components in the simulation chain, i.e. not only the flow solver, but also the parametrisation. Gradient computation can be easily implemented using finite differences, however this incurs truncation or round-off errors that can be difficult to control. Alternatives are the complex step method (Squire and Trapp 1998) or algorithmic differentiation (Griewank and Walther 2008; Naumann 2011) which can produce exact derivatives, but the computational cost scales linearly with the number of design variables. The adjoint method allows to compute the gradients of an objective with respect to an arbitrary number of design variables at constant cost and has hence become the method of choice in CFD (Pironneau 1974; Jameson 1988). However, implementing an adjoint solver is not trivial, both the continuous approach (Jameson 1995) and the discrete one (Giles and Duta 2003; Jones et al. 2011; Christakopoulos et al. 2011) require a significant effort.

One of the key issues in shape optimisation is the parametrisation of the geometry, because it will determine the design space and thus the optimisation result. Current parametrisation methods can roughly be distinguished as CAD-free and CAD-based ones. In the CAD-free approaches, the parametrisation has no link back to the CAD geometry, which makes these approaches easy to implement. Examples are mesh deformation through Radial Basis Functions (Jakobsson and Amoignon 2007), Free-form Deformation with volume splines (Samareh 2004) and node-based approaches (Jameson and Vassberg 2000; Stephan et al. 2008; Jaworski and Müller 2008). The downside of these approaches is that the optimal result needs to be transcribed back to CAD which is often cumbersome and may lose relevant features of the optimised shape.

In CAD-based methods, the parametrisation of the shape is part of the CAD model which is kept inside the design loop. Hence, the resulting optimal shape is directly available for further analysis or manufacturing. The main difficulty for gradient-based optimisation is that we also need to compute derivatives of the parametric CAD model.

The parametrisation can be defined *explicitly* in a parametric CAD system, as often done to generate a family of parts in different sizes or for manual design space exploration. While this approach has the advantage that geometric constraints such as thicknesses or radii can be directly built into the design space, typically these parametrisations either do not have sufficient freedom to capture relevant modes, or require important human knowledge about the flow and incur significant user effort to set up, which ultimately negates the benefits of numerical optimisation. Derivative computation of the explicitly parametrised CAD model can then either through finite-differencing, which incurs the typical problems of finite differences with truncation errors and choice of step size. Robinson et al. (2009) avoid problems of robustness

due to patch renumbering by approximating the surfaces with triangulations (STL), which in turn has issues with projection near sharp corners.

Gradients can also be computed by applying automatic differentiation to the source code of the CAD system, which addresses issues of accuracy and will allow use of the efficient reverse mode (Auriemma et al. 2016). This approach still requires to define a suitable CAD parametrisation.

Alternatively, a CAD-based parametrisation can also be defined *implicitly*, i.e. to arise from the CAD model's generic description such as the collection of NURBS patches in the STEP or IGES standards. The positions and weights of the control points (CP) of the NURBS patches can be used as design variables.

In this work, the QMUL in-house CAD kernel, termed NURBS-based parametrisation with continuity constraint (NSPCC) (Xu et al. 2014; Zhang et al. 2016) is used to parametrise the geometry, and is integrated into the CAD-based optimisation loop. This approach offers a number of advantages. Firstly only a subset of CAD functionality is needed which can straightforwardly be implemented in light-weight standalone tools, which in turn significantly simplifies the derivative computation with automatic differentiation tools (Xu et al. 2014). Secondly the implicit parametrisation through control points typically produces a suitably rich design space without need for manual setup (Jesudasan et al. 2016). On the downside, a methodology needs to be introduced for imposition of geometric constraints, such as continuity constraints between NURBS patches or thickness, box and radii constraints. This can be achieved effectively with a test-point methodology (Xu et al. 2014, 2015). The design space is in most cases excessively rich, however when using the adjoint approach to evaluate the gradients, this is not an inconvenience. Provided that the design space remains coarser than the CFD mesh, gradient regularisation is not required, as opposed to mesh-based parametrisations (Jameson and Vassberg 2000). However, rich design spaces may lead to shapes with mildly oscillatory modes that are not detrimental to the objective function, hence not penalised by the flow solver, but may be undesirable as they are visually unappealing or affect another discipline/behaviour that is not modelled. This aspect is addressed in this paper.

In this paper we will present a methodology that uses as free parameters not only the positions of the NURBS control points, but also their weights. This enables to accurately represent shapes with coarser control lattices, resulting in smoother changes of curvature which can be important in the design of turbo-machinery blades or transonic wings.

The remaining part of this paper is structured as follows: Section “[Adjoint Approach](#)” introduces the adjoint approach briefly, then Section “[Parametrisation](#)” provides information on NURBS and also the NSPCC approach. The approximation of ONERA M6 wing using NURBS will be presented in Section “[M6 Wing Profile Approximation Using NURBS](#)”, followed by the optimisation results in Section “[Drag Minimisation of the M6 Wing](#)”. Finally, conclusions will be given in Section “[Conclusions](#)”.

Adjoint Approach

Let us consider the Navier-Stokes equations as

$$\mathbf{R}(\mathbf{U}, \boldsymbol{\alpha}) = 0, \quad (1)$$

where \mathbf{R} is the conservative residual of flow equations, \mathbf{U} is state variables and $\boldsymbol{\alpha}$ is a set of design variables. Taking the derivative of (1) with respect to $\boldsymbol{\alpha}$, we have:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} + \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} = 0, \quad (2)$$

which can be written as

$$\mathbf{A} \mathbf{u} = \mathbf{f}, \quad (3)$$

where \mathbf{A} is the Jacobian $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}$, \mathbf{u} is the perturbation field, i.e. the change of the flow field with respect to $\boldsymbol{\alpha}$ and \mathbf{f} is the change in residual w.r.t. changes in shape, $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}$.

The sensitivity of the objective function J with respect to the design variables $\boldsymbol{\alpha}$ can then be formulated as:

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \frac{\partial J}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \mathbf{g}^T \mathbf{u}, \quad (4)$$

where $\mathbf{g}^T = \frac{\partial J}{\partial \mathbf{U}}$. In (4), the computation of the partial derivative $\frac{\partial J}{\partial \boldsymbol{\alpha}}$ is not expensive. Similarly, the source term f in (3) is computationally inexpensive, but depends on the design variable α_i . However, computing \mathbf{u} involves solving a linear system solve for each component α_i of $\boldsymbol{\alpha}$, a cost which will become prohibitive if the number of design variables is large.

On the other hand, the sensitivity of the objective $\frac{dJ}{d\boldsymbol{\alpha}}$ can be obtained without computing the perturbation field \mathbf{u} with the adjoint method. Using the solution of (3), $\mathbf{u} = \mathbf{A}^{-1} \mathbf{f}$, in (4) and transposing one obtains

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + (\mathbf{g}^T \mathbf{A}^{-1} \mathbf{f})^T = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \mathbf{f}^T \mathbf{A}^{-T} \mathbf{g}. \quad (5)$$

The right-most matrix-vector product $\mathbf{A}^{-T} \mathbf{g}$ can be interpreted as a linear equation similar to equation (3) for a new variable \mathbf{v} ,

$$\mathbf{A}^T \mathbf{v} = \mathbf{g}. \quad (6)$$

With a solution for the adjoint variable \mathbf{v} we can rewrite (5) as

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{\partial J}{\partial \boldsymbol{\alpha}} + \mathbf{v}^T \mathbf{f}. \quad (7)$$

The adjoint equivalence then states

$$\mathbf{g}^T \mathbf{u} = (\mathbf{A}^T \mathbf{v})^T \mathbf{u} = \mathbf{v}^T \mathbf{A} \mathbf{u} = \mathbf{v}^T \mathbf{f}. \quad (8)$$

Since the adjoint variable \mathbf{v} depends only on the flow field through the transposed Jacobian \mathbf{A}^T and on the objective through \mathbf{g} , (7) allows to compute the entire sensitivity vector $\frac{dJ}{d\boldsymbol{\alpha}}$ with a single system solve of (6). Hence, the cost of computing the gradient becomes independent of the number of design variables.

Using the chain rule of calculus, the differentiation of the objective function can be broken down into a number of steps, aligned with the computational workflow,

$$\frac{\partial J}{\partial \boldsymbol{\alpha}} = \frac{\partial J}{\partial x_V} \frac{\partial x_V}{\partial x_s} \frac{\partial x_s}{\partial \boldsymbol{\alpha}}, \quad (9)$$

with the volume grid coordinates x_V , which in turn are linked to perturbations of the surface grid coordinate x_s through a mesh deformation algorithm, which in turn are affected by modifications of the CAD parameters $\boldsymbol{\alpha}$. The first term on the right hand side $\partial J / \partial x_V$ is computed by differentiating the flow solver, the second term $\partial x_V / \partial x_s$ by differentiating the the volume mesh smoothing, while the third term $\partial x_s / \partial \boldsymbol{\alpha}$ requires a differentiation of the CAD model parametrisation.

In our work we employ Automatic Differentiation (AD) Software tools Tapenade (Hascoët and Pascual 2004) to produce the adjoint code and assemble the routines in a hand-written driver code to improve performance (Christakopoulos et al. 2011).

Parametrisation

Parametrisation of geometry is crucial in shape optimisation as it will affect the design space. There are various parametrisation methods which can generally be distinguished as CAD-free and CAD-based methods. In this work, we focus on the CAD-based approach using boundary representation (BRep), where NURBS is the standard way to express geometries.

NURBS Surface Patches

Non-Uniform Rational B-splines (NURBS) are widely used to describe geometries. A NURBS patch is a 3D surface defined as (Piegl and Tiller 1997):

$$\mathbf{S}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \omega_{i,j}} \quad 0 \leq u, v \leq 1, \quad (10)$$

where $\mathbf{P}_{i,j}$ are the control point coordinates, $\omega_{i,j}$ their corresponding weights, $N_{i,p}(u)$ and $N_{j,q}(v)$ the p -th and q -th degree B-spline basis functions defined in the following knot vectors:

$$\underbrace{\{0, \dots, 0, u_{p+1}, \dots, u_i, \dots, u_{r-p-1}, 1, \dots, 1\}}_{p+1}$$

$$\underbrace{\{0, \dots, 0, v_{q+1}, \dots, v_j, \dots, v_{s-q-1}, 1, \dots, 1\}}_{q+1}$$

where $r = n + p + 1$ and $s = m + q + 1$. $N_{i,p}(u)$ and $N_{j,q}(v)$ are given by the following expression:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,k}(u) = \frac{(u - u_i)}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{(u_{i+k+1} - u)}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u). \quad (11)$$

NURBS hold many geometric properties, making them very suitable to describe geometries in shape optimisation problems. Some of them are:

- Local modification. If a control point is perturbed, only a part of the geometry will be affected.
- Generalisation. NURBS can be used to describe very wide range of geometries, including circular and conic shapes, which the B-splines can not express exactly.
- Strong convex hull property.
- Affine invariance.

A NURBS surface can be expressed in the so-called homogeneous form:

$$\mathbf{S}^\omega(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j}^\omega \quad 0 \leq u, v \leq 1, \quad (12)$$

where $\mathbf{P}_{i,j}^\omega = (\omega_{i,j} x_{i,j}, \omega_{i,j} y_{i,j}, \omega_{i,j} z_{i,j}, \omega_{i,j})$. This homogeneous form is very similar to a B-spline surface, therefore if written in this form, most of the algorithms for B-spline surfaces can straightforwardly be applied to NURBS. In this way, the weights can also be used in shape optimisation problems.

NSPCC Approach

NSPCC is an in-house lightweight CAD kernel, which uses an implementation in Fortran 90 of NURBS patches which can then be differentiated using the source-transformation AD tool Tapenade (Hascoët and Pascual 2013). Xu et al. (2014)

considered B-splines and used the coordinates of the control points \mathbf{P}_{ij} to derive the design variables. Zhang et al. (2016) extended NSPCC from B-splines to NURBS and also include the weights ω_{ij} in the design space.

The important contribution of NSPCC to CAD-based parametrisation based on the BRep is the formulation of geometric constraints, e.g. G0-G2 continuity at NURBS patch interfaces. The constraint equations are formulated numerically at a sufficiently large number of test-points (Zhang et al. 2016) and the derivatives of these equations are computed using AD. The design space is the kernel of this matrix of constraint derivatives and is evaluated using singular value decomposition (SVD). The design variables are ultimately the linear combination coefficients of the SVD basis for the kernel. Details of the NSPCC approach can be found in Xu et al. (2014, 2015). Using the homogeneous form (12) then allows to extend the constraint framework straightforwardly from B-Splines to NURBS.

The scale and effect of the design variables will have a significant influence on the convergence toward the optimum. Considering e.g. a curve aligned with the x -axis. Control point movements in the curve-normal y direction will have a much stronger effect on the shape than movements in the tangential x -direction. Similarly, the weights have scalings around unity, while the scaling of the coordinates entirely depends on the measurement units used for the coordinate values. In the NSPCC approach, the scaling strategy from Painchaud-Ouellet et al. (2006) is applied to control points such that a variation of the scaled control points in x direction between ± 1 causes around $\pm 10\%$ variation of the root chord length, and a variation of scaled control points in y direction yields $\pm 10\%$ deformation of the maximum thickness of root profile.

NSPCC uses the STEP file (Pratt 2001) as input and output for the geometry. The algorithm is hence independent of any internal proprietary parametrisation inside a particular CAD system.

M6 Wing Profile Approximation Using NURBS

The M6 wing (Schmitt and Charpin 1979) is a typical example of a geometry that cannot be represented exactly in CAD, hence defining a CAD model for the M6 requires a trade-off between fidelity and complexity. A similar case is the approximation of a cylindrical shape with B-splines, the standard choice for approximation in CAD systems. A low tolerance of deviation between B-spline surface and the analytic cylindrical shape will result in a B-spline approximation with very many control points. On the other hand, a NURBS-representation allows to match a cylinder exactly with 6 distinct control points for a circular section. Lépine et al. (2001) fitted NURBS to reduce the number of control points for 2D aerofoils. In this work, we directly approximate the 3D wing shape with NURBS in the NSPCC framework. This has two major advantages. Firstly, even though the cost of computing the derivatives is constant with the adjoint approach, the lower number of control points will make it easier for optimiser to converge the KKT system. Secondly, and more

importantly, the smaller number of control points will result in a smoother variation of curvature along the profile, which is an essential quality in the design of transonic wings and turbomachinery blades.

As a first step, we compare the approximation of datum M6 wing shape using B-Spline and NURBS representations. The M6 wing created with 26 B-splines control points for each section (see Fig. 3) (Gugala et al. 2014; Gugala 2017) according to the description in Schmitt and Charpin (1979) is used as the target shape. The wing parametrised with different numbers of NURBS control points are used as starting point. Then the cost function of this optimisation problem is defined as the root-mean-squared difference between initial and target geometries, i.e.:

$$J = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{X}_{Initial_i} - \mathbf{X}_{Target_i})^2}, \quad (13)$$

where $\mathbf{X}_{Initial_i}$ and \mathbf{X}_{Target_i} are surface points sampled on the initial and target wing geometry, respectively. N is the number of sample points. To improve the surface fidelity in regions of high curvature, a cosine spacing is used which concentrates the sampling points near the leading edge than near the trailing edge, as shown in Fig. 1.

For the M6 fitting in this study, 5000 surface points are sampled on each geometry, 100 points in the chordwise and 50 points along the spanwise direction.

The initial wing geometries with different number of NURBS control points are driven towards the target shape by performing optimisation using the L-BFGS method (Nocedal 1980; Liu and Nocedal 1989). The derivative of the objective function w.r.t. the design variables, i.e. the 2-D coordinates of the control points in the profile plane and their weights, are calculated using AD.

The objective function values (13) resulting from the optimisation with varying numbers of control points is given in Fig. 2. It can be observed that the objective value continues to drop with increased number of control points. We use the typically accepted value of 10^{-4} to determine an acceptable fit.

The M6 wing described using 26 B-spline control points and 16 NURBS control points are presented in Fig. 3. The comparison of one section is shown in Fig. 4.

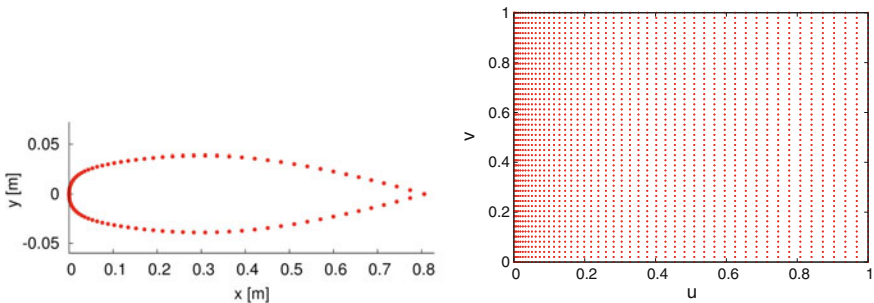


Fig. 1 Cosine spacing sample points. Left: points of one section. Right: points in parametric space

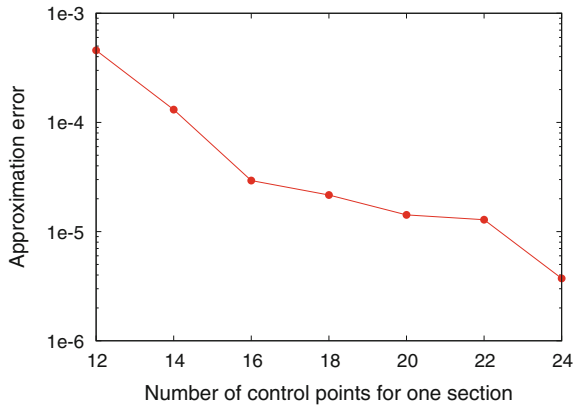


Fig. 2 Fidelity of NURBS approximation with varying number of control points

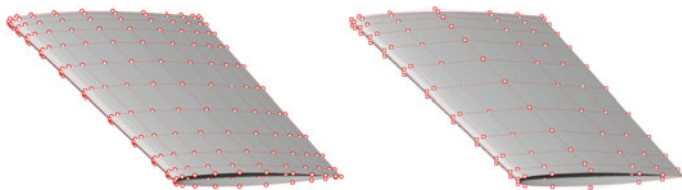


Fig. 3 ONERA M6 wing with 26 B-spline (left) and 16 NURBS control points (right)

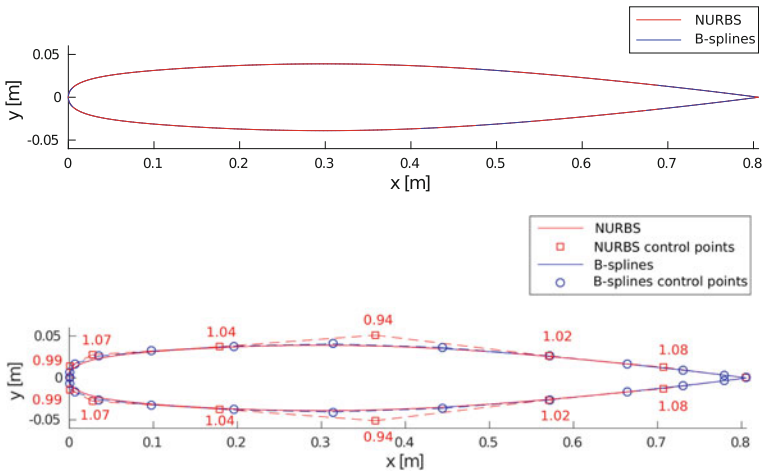


Fig. 4 Comparison of ONERA M6 wing root section using 26 B-spline and 16 NURBS control points. Upper: shape of section. Lower: shape of section, control points and optimised weights

The results clearly demonstrated that by using NURBS, the number of control points required to describe the M6 wing with a given surface and curvature fidelity is significantly reduced.

Drag Minimisation of the M6 Wing

In-House Solver: STAMPS

The in-house solver STAMPS (Müller et al. 2016; Gugala 2017) (Source-Transformation Adjoint Multi-physics Solver) is used to perform flow analysis and provide the sensitivity of the objective function w.r.t. surface mesh node positions, $\partial J / \partial x_V$. STAMPS is a compressible flow solver using a vertex-centred Finite Volume Method (FVM). A discrete adjoint solver which is derived using the AD tool Tapenade is also included in STAMPS to provide sensitivities (Christakopoulos et al. 2011).

Case Set Up

The optimisation of the ONERA M6 wing in the transonic regime and inviscid flow are performed. The key case information are listed as following:

- Mesh: tetrahedron, 135204 nodes.
- Flow conditions: $Ma = 0.84$, Angle of attack (AoA) = 3.06° , $T_\infty = 300K$, $p_\infty = 101325 Pa$.
- Mesh deformation method: inverse distance weighting (IDW) (Witteveen and Bijl 2009).
- Control points at the leading and trailing edge are frozen to fix the AoA. Other control points can move vertically and along the chordwise direction. Weights of free control points can also change.
- Cost function: drag with lift constraints, defined as:

$$J = D + c * (L - L^*)^2, \quad (14)$$

where D is the drag force, c is the penalty coefficient chosen as 0.0001, L is lift force, and L^* is the initial lift, which is 11104.21 N here.

The pressure contours of the baseline geometry are shown in Fig. 5. As it can be seen, the typical ‘lambda’ shock on the top surface is very clear.

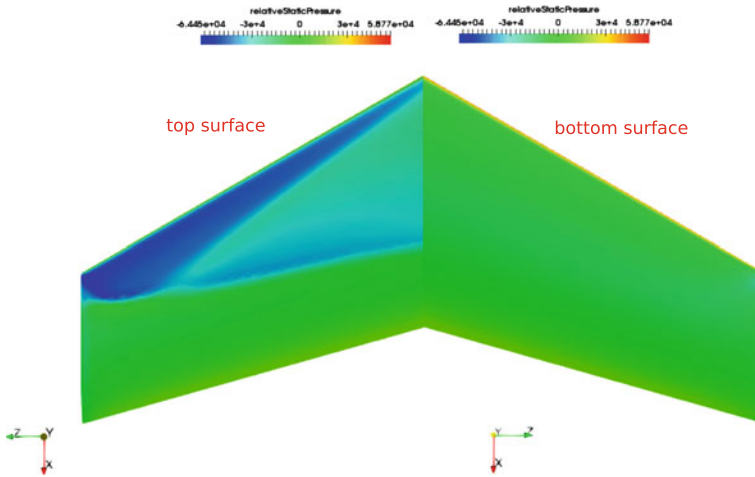


Fig. 5 Baseline geometry pressure contours on top and bottom surface

Optimisation Results

To investigate the effect of parametrisation on optimisation results, the M6 wing described using 26 B-splines and 16 NURBS control points for each section are used in the optimisation. The steepest descent method with Armijo line search is utilised as optimiser (Bartholomew-Biggs 2008).

The comparison of drag, lift, efficiency (lift/drag ratio) and objective function during the optimisation are shown in Fig. 6. In both cases, the drag values are reduced by over 30%. In the NURBS case, the optimised wing loses 1.64% of lift, which is slightly better than of B-spline case (1.98%). The loss of some lift is as expected, because the penalty part in the objective function is not very strict such that more drag

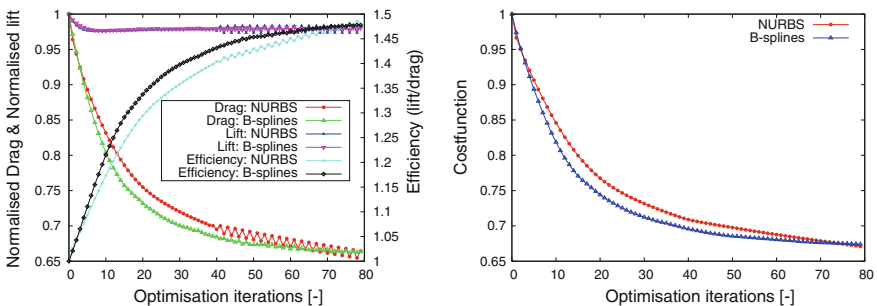
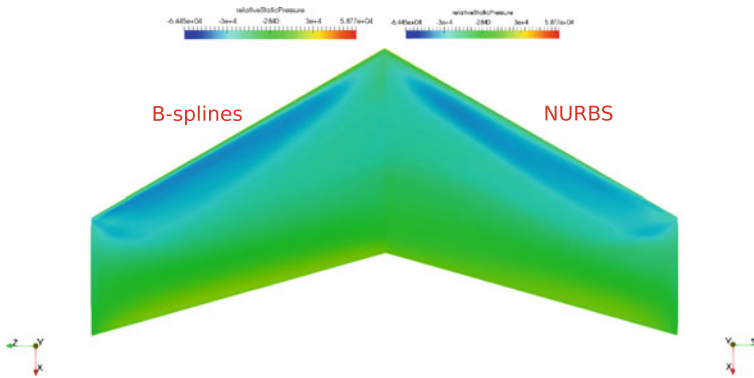


Fig. 6 Left: The comparison of normalised drag, lift and efficiency (lift/drag ratio). Right: The comparison of cost function

Table 1 Final drag and lift coefficients

	Drag coefficient	Lift coefficient	Objective function (scaled)
B-splines	0.00850	0.2750	0.00 864
NURBS	0.00850	0.2759	0.00860

**Fig. 7** Comparison of pressure contour on upper surface after optimisation

reduction can be achieved. Exact values of final drag and lift are listed in Table 1. Note that the NURBS case optimisation has been stopped at the same number of iterations as the B-spline case, however the gradient values of the NURBS case are still much larger and further improvement with more design iterations can be expected.

The comparison of pressure contours on upper surface after optimisation is presented in Fig. 7, which clearly indicates that in both cases the shocks on the upper surface are reduced significantly. This is also well illustrated in Fig. 8 where the pressure coefficients (C_p) on both upper and lower surface at different spanwise positions are given.

The comparison of wing profiles at different spanwise positions are given in Fig. 8. One can see that the optimised shape of in the B-splines case has a large curvature variation near the trailing edge. A clearer comparison of curvature is presented in Figs. 9 and 10, which show the curvature of the two wing sections and the mean curvature (Spivak 1999) of optimised surfaces, respectively. These figures clearly demonstrate that the optimisation using a NURBS parametrisation produces a smoother shape. The possible reason is that, when fewer control points are used, the possibility of producing noisy surface is reduced. As a consequence, the aerodynamic performance is better, since in the transonic flow the aerodynamic performance is very sensitive to the shape of geometry. It is expected that when fully converged, the

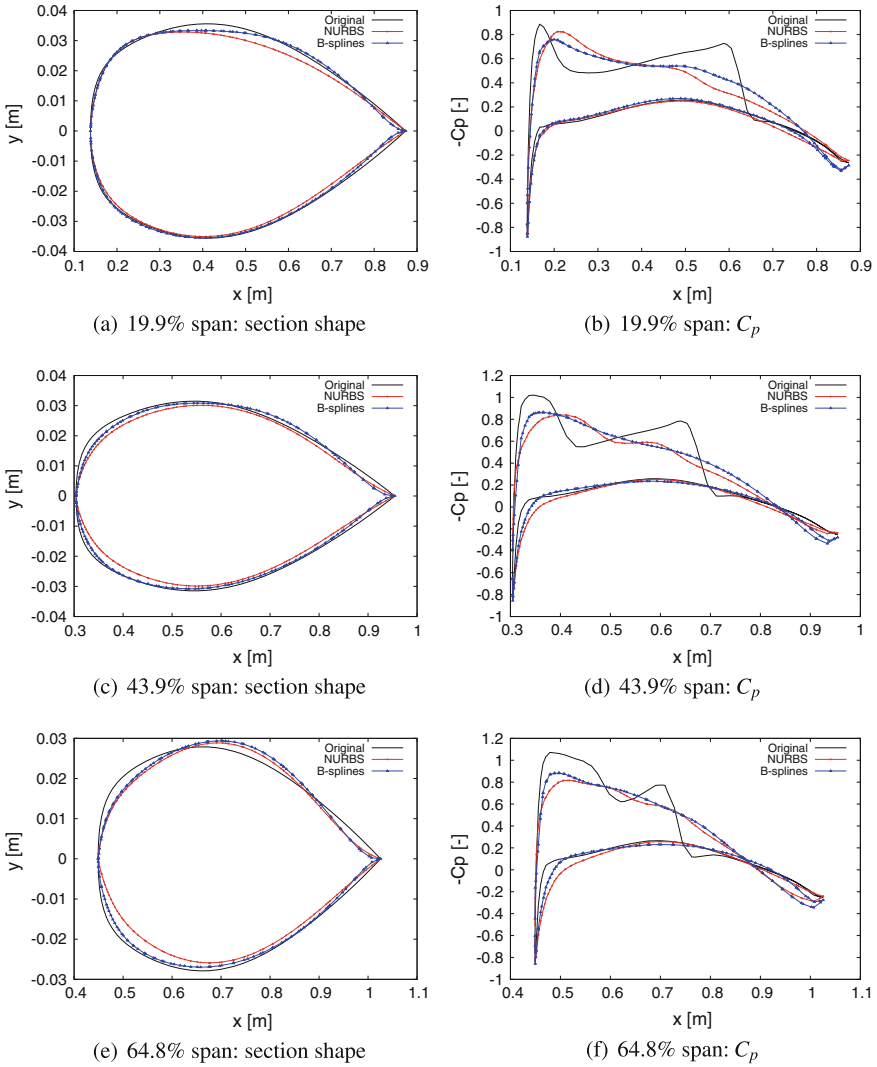


Fig. 8 Comparison of cross section and pressure coefficient at different spanwise position

NURBS parametrisation will produce a lower drag since the much smoother variation of curvature. The smoother curvature variation will be more important in viscous flow, as strong changes in curvature will lead to rapid pressure changes which will adversely affect the boundary layer.

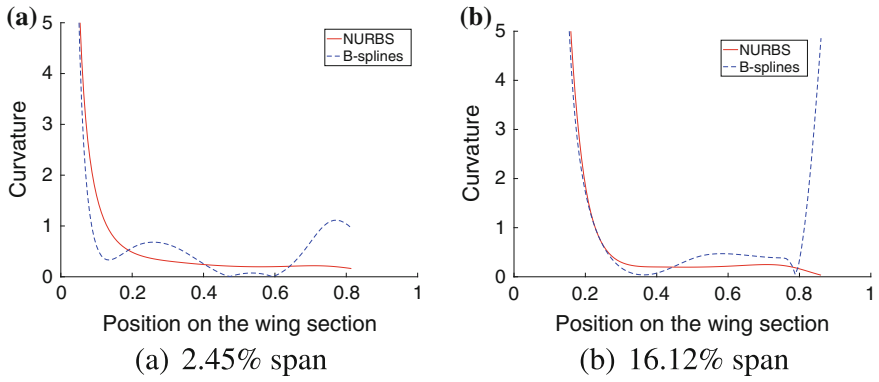


Fig. 9 Comparison of curvature along wing sections

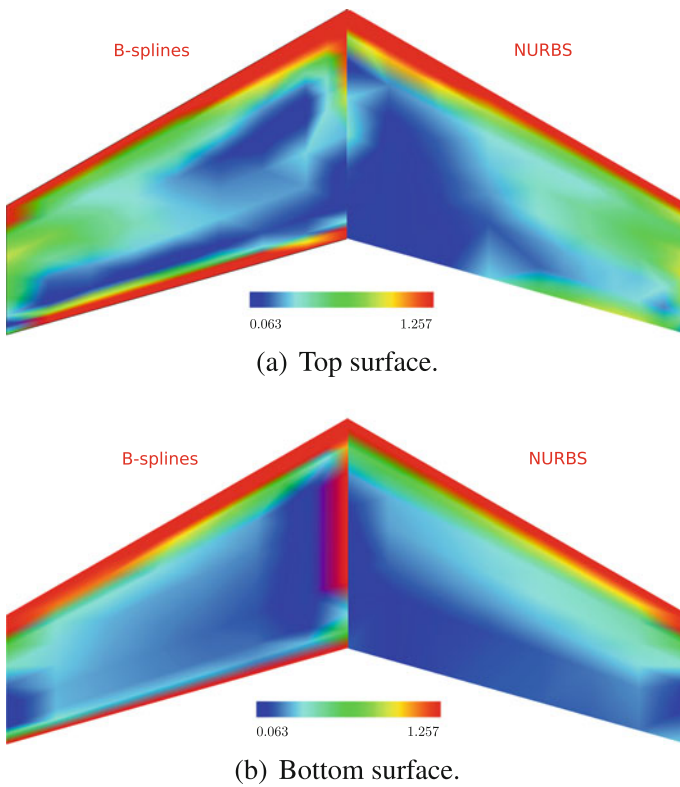


Fig. 10 Comparison of mean curvature of optimal wing surfaces

Conclusions

In this paper, NURBS have been demonstrated as an appropriate parametrisation method for wing shape optimisation. NURBS associate a weight with each control point which is additional freedom in controlling shapes, and have hence been shown to be capable to represent shapes accurately with fewer control points. This helps to produce surface with smaller curvature variation. The optimisation results of ONERA M6 wing with both B-splines and NURBS parametrisation in transonic inviscid flow based on adjoint method have been presented. It has been shown that the optimisation using a NURBS-based wing parametrisation has a smoother shape with smaller variation of curvature, which is beneficial for aerodynamic performance. This can be important in the design of turbo-machinery blades or transonic wings.

Finally, the effectiveness of CAD-based optimisation coupling the lightweight NSPCC approach with adjoint method has also been demonstrated.

Acknowledgements The first author would like to thank the China Scholarship Council (No. 201306230097) and Queen Mary University of London for funding this research. The second author acknowledges the support from the IODA project (<http://ioda.sems.qmul.ac.uk>), funded by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

References

- Auriemma S, Banovic M, Mykhaskiv O, Legrand H, Müller JD, Walther A (2016) Optimisation of a U-bend using CAD-based adjoint method with differentiated cad kernel. In: ECCOMAS Congress
- Bartholomew-Biggs M (2008) Nonlinear optimization with engineering applications, vol 19. Springer, Berlin
- Christakopoulos F, Jones D, Müller J-D (2011) Pseudo-timestepping and validation for automatic differentiation derived code. *Comput Fluids* 46(1):174–179
- Giles MB, Duta MC, Müller J-D, Pierce NA (2003) Algorithm developments for discrete adjoint methods. *AIAA J* 41(2):198–205
- Griewank A, Walther A (2008) Evaluating derivatives: principles and techniques of algorithmic differentiation. PA, 2nd edn, SIAM, Philadelphia
- Gugala M (2017) Output-based mesh adaptation using geometric multi-grid for error estimation. PhD thesis, Queen Mary University of London
- Gugala M, Xu S, Mueller J-D (2014) Node-based vs CAD-based Approach in CFD Adjoint- based Shape Optimisation. In: International Conference on Engineering and Applied Sciences Optimization, Kos, Greece
- Hascoët L, Pascual V (2004) Tapenade 2.1 user's guide. Technical Report 0300, INRIA
- Hascoët L, Pascual V (2013) The Tapenade Automatic Differentiation tool: principles, Model, and Specification. *ACM Trans Math Softw* 39(3):
- Jakobsson S, Amoignon O (2007) Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Comput Fluids* 36(6):1119–1136
- Jameson A (1988) Aerodynamic design via control theory. *J Sci Comput* 3:233–260
- Jameson A (1995) Optimum aerodynamic design using CFD and control theory. *AIAA paper* 1729:124–131

- Jameson A, Vassberg A (2000) Studies of alternative numerical optimization methods applied to the Brachistochrone problem. *Comput Fluid Dyn J* 9(3):
- Jaworski A, Müller J-D (2008) Toward modular multigrid design optimisation. In: Bischof C, Utke J (eds) *Lecture notes in computational science and engineering*, vol 64. Springer, New York, pp 281–291
- Jesudasan R, Zhang X, Gugala M, Mueller JD (2016) CAD-free vs CAD-based parametrisation method in adjoint-based aerodynamic shape optimization. In: *ECCOMAS Congress 2016*
- Jones D, Christakopoulos F, Müller J-D (2011) Preparation and assembly of adjoint CFD codes. *Comput Fluids* 46(1):282–286
- Lépine J, Guibault F, Trépanier J-Y, Pépin F (2001) Optimized nonuniform rational B-spline geometrical representation for aerodynamic design of wings. *AIAA J* 39(11):2033–2041
- Liu D, Nocedal J (1989) On the limited memory bfgs method for large scale optimization. *Math Program* 45(1):503–528
- Müller J-D, Gugala M, Xu S, Hüchelheim J, Mohanamuraly P, Imam-Lawal OR (2016) Introducing STAMPS: an open-source discrete adjoint CFD solver using source-transformation ad. In: *11th ASMO UK/ISSMO/NOED2016: International Conference on Numerical Optimisation Methods for Engineering Design*
- Naumann U (2011) *The art of differentiating computer programs: an introduction to algorithmic differentiation*. SIAM
- Nocedal J (1980) Updating quasi-Newton matrices with limited storage. *Math Comput* 35(151):773–782
- Painchaud-Ouellet Simon, Tribes Christophe, Trépanier Jean-Yves, Pelletier Dominique (2006) Airfoil shape optimization using a nonuniform rational B-splines parametrization under thickness constraint. *AIAA J* 44(10):2170–2178
- Piegl L, Tiller W (1997) *The NURBS Book*. 2nd edn
- Pironneau O (1974) On optimum design in fluid mechanics. *J Fluid Mech* 64:97–110
- Pratt MJ (2001) Introduction to ISO 10303—the STEP standard for product data exchange. *J Comput Inf Sci Eng* 1(1):102
- Robinson TT, Armstrong CG, Chua HS, Othmer C, Grahs T (2009) Sensitivity-based optimization of parameterised CAD geometries. In: *8th World Congress on Structural and Multidisciplinary Optimisation*, Lisbon
- Samareh JA (2004) Aerodynamic shape optimization based on free-form deformation. *AIAA*, paper 4630:1–13. <https://doi.org/10.2514/6.2004-4630>
- Schmidt S, Ilic C, Gauger N, Schulz V (2008) Shape gradients and their smoothness for practical aerodynamic design optimization. DFG SFB 1253 Preprint-Number SPP1253-10-03, Universität Erlangen
- Schmitt V, Charpin F (1979) Pressure distributions on the Onera-M6-wing at transonic mach numbers. *Experimental data base for computer program assessment*, 4
- Spivak M (1999) *A comprehensive introduction to differential geometry*. Publish or perish, Houston
- Squire W, Trapp G (1998) Using complex variables to estimate derivatives of real functions. *SIAM-Review* 10(1):110–112
- Witteveen JAS, Bijl H (2009) Explicit mesh deformation using inverse distance weighting interpolation. In: *19th AIAA Computational Fluid Dynamics*, p 3996
- Xu S, Jahn W, Müller J-D (2014) CAD-based shape optimisation with CFD using a discrete adjoint. *Int J Numer Methods Fluids* 74(3):153–168
- Xu S, Radford D, Meyer M, Müller J-D (2015) CAD-based adjoint shape optimisation of a one-stage turbine with geometric constraints. *ASME Turbo Expo 2015, GT2015-42237*
- Zhang X, Wang Y, Gugala M, Müller J-D (2016) Geometric continuity constraints for adjacent NURBS patches in shape optimisation. In: *ECCOMAS Congress 2*

Part II
Surrogate-Assisted Optimization of Real
World Problems

A Comparative Evaluation of Surrogate Models for Transonic Wing Shape Optimization



Emiliano Iuliano

Abstract The paper details a comparative analysis of different models able to provide a fast response within a surrogate-based shape optimization process. Kriging, Radial Basis Function Network (RBFN) and Proper Orthogonal Decomposition in combination with RBFNs (POD+RBFN) are employed as fitness function evaluators within the framework of evolutionary algorithms (EAs). The surrogate-assisted optimization consists of initializing the surrogate with space-filling samples, improving the accuracy by adding a series of “smart” samples through specifically designed in-fill criteria and finally optimizing on the surrogate. The test case is represented by the large scale shape optimization of a transonic wing in viscous flow and in multi-design point conditions. Optimization results obtained with the surrogates by fixing the total computational budget are presented: this procedure allows to make a fair comparison between the models and their performance during the optimization process.

Introduction

In real-world engineering design applications, high-fidelity simulations and reliable answers in short time are essential and fundamental requirements. Of course, they are often conflicting, especially when fluid dynamics is among the physical disciplines to be solved: indeed, computational fluid dynamics (CFD) simulations of complex configurations are still time-consuming and, considering also the high number of CFD simulations required by global optimization approaches, this strongly hampers the usage of such methods in engineering design.

Surrogate-based optimization (SBO) may provide an interesting answer to this issue as it relies on a fast response model to be used during optimization while

E. Iuliano (✉)

Fluid Mechanics Department, Multidisciplinary Analysis and Optimization Group,
Centro Italiano Ricerche Aerospaziali (CIRA),
Via Maiorise, 81043 Capua, Italy
e-mail: e.iuliano@cira.it

invoking the “truth” model (i.e., the CFD simulation) to confirm the choice made by the surrogate. Several researchers have focused their attention on such topic, both from a theoretical (Forrester and Keane 2009; Braconnier et al. 2011; Viana et al. 2012) and application (Robinson et al. 2006; Booker et al. 1999; Mack et al. 2007) point of view. As a consequence, a varied amount of methods exist which differ substantially for the choice of the surrogate model (e.g., type, single or multiple), the approach to build the surrogate (e.g., optimize the generalization error or likelihood functions), the strategy for updating and improving the surrogate (e.g., evaluate surrogate minimizers, use in-fill criteria, random choice) and the optimization method (e.g., type, global or local or both).

The present paper proposes different choices of the surrogate model to be used within a SBO cycle with different updating strategies. The problem at hand is the multi-point shape optimization of a wing in viscous transonic conditions: such a problem stems as a large-scale and real-world optimization as it involves several design parameters and black-box CFD-based functions. As a consequence, in principle it cannot be handled by whatever methodology and the main aim is to provide arguments in support of the successful usage of accurate “optimal” surrogates and global optimization techniques.

Surrogate Models

This section is devoted to introduce the mathematical basis of the meta-models which will be used for surrogate-based optimization. Kriging and Radial Basis Function Network models work with scalar information (e.g., the objective function values) and are able to predict the response function at each location of the design space. On the other hand, the Proper Orthogonal Decomposition is coupled to Radial Basis Function Network models to deal with vector quantities (e.g., the flow field) and, thus, to inject more physics information within the surrogate training process.

Kriging

The Kriging model is built on the assumption that the training data obey a Gaussian process with an assumed form for the mean function and the covariance between data points. A Kriging surrogate models the response of interest $f(\mathbf{x})$ as a realization of a regression model h and a stochastic process z (Martin and Simpson 2005):

$$f(\mathbf{x}) = h(\beta, \mathbf{x}) + z(\mathbf{x}) \quad (1)$$

$$h(\beta, \mathbf{x}) = \mathbf{h}\beta \quad (2)$$

$$\mathbb{E}[z(\mathbf{x}_1), z(\mathbf{x}_2)] = \sigma_k^2 R(\theta, \mathbf{x}_1, \mathbf{x}_2) \quad (3)$$

where β are the regression coefficients and \mathbf{h} is the regression vector. The stochastic process z is assumed to have zero mean, process variance σ_k^2 and covariance model $R(\theta, \mathbf{x}_1, \mathbf{x}_2)$ between $z(\mathbf{x}_1)$ and $z(\mathbf{x}_2)$ with parameters vector θ . The covariance model between function values is assumed to be only a function of the distance between points. Given the training sites $\{\mathbf{x}_j\}_{j=1,\dots,M}$, the covariance matrix is given by $K_{ij} = R(\theta, \mathbf{x}_i, \mathbf{x}_j)$. Multi-dimensional covariance is built up using a tensor product of one-dimensional covariance functions:

$$R(\theta, \mathbf{x}_i, \mathbf{x}_j) = \prod_p^D Kr \left(\left| \frac{x_{ip} - x_{jp}}{\theta_p} \right| \right)$$

where D is the dimension of the problem, θ_p is the length scale in the p -th dimension, x_{ip} is the p -th component of the vector \mathbf{x}_i and Kr is the one-dimensional Matern function. The latter function is computed as:

$$Kr(d) = \exp(-\sqrt{2\nu}d) \frac{\Gamma(t+1)}{\Gamma(2t+1)} \sum_{i=0}^t \frac{(t+i)!}{i!(t-i)!} (\sqrt{8\nu}d)^{t-i}$$

with Γ the Gamma function, $\nu = t + 1/2$ and three possible values of the parameter t :

$$Kr(d) = \begin{cases} \exp(-d) & \text{for } t = 0 \\ (1 + \sqrt{3}d) \exp(-\sqrt{3}d) & \text{for } t = 1 \\ (1 + \sqrt{5}d + \frac{5}{3}d^2) \exp(-\sqrt{5}d) & \text{for } t = 2 \end{cases}$$

Noise terms can be added along the covariance matrix diagonal in order to improve the matrix conditioning and to obtain a regressive behavior when dealing with noisy functions. The covariance matrix becomes $K_{ij} = R(\theta, \mathbf{x}_i, \mathbf{x}_j) + \lambda\delta_{ij}$, where the Kronecker convention has been used and λ is the noise ratio. The response function can be estimated at a generic location \mathbf{x} as

$$\hat{f}(\mathbf{x}) = \mathbf{H}\hat{\beta} + \mathbf{k}^T \mathbf{K}^{-1}(\mathbf{f} - \mathbf{H}\hat{\beta}) \tag{4}$$

where \mathbf{H} is the matrix of linear equations constructed using the regression function and the training sites, $\hat{\beta}$ is the generalized least square estimate of β , \mathbf{K} is the covariance matrix, \mathbf{k} is the covariance vector between the generic design site \mathbf{x} and the training sites, and $\mathbf{f} = [f_1, f_2, \dots, f_M]^T$ is the vector of the M training data (which corresponds to the given training dataset). One of the main advantages of the Kriging model is that it provides also an estimate of the prediction variance:

$$\hat{s}^2(\mathbf{x}) = \hat{\sigma}_k^2 [1 - \mathbf{k}^T \mathbf{K}^{-1} \mathbf{k} + \mathbf{u}^T (\mathbf{H}^T \mathbf{K}^{-1} \mathbf{H})^{-1} \mathbf{u}] \tag{5}$$

where $\hat{\sigma}_k^2$ is the estimated process variance, $\mathbf{u} = \mathbf{H}^T \mathbf{K}^{-1} \mathbf{k} - \mathbf{h}$ and $\mathbf{h} = [h_1, h_2, \dots, h_M]^T$. In the most general case, both the response prediction \hat{f} and the prediction variance \hat{s}^2 are function of the so called hyperparameters, i.e. the length scales θ_p , the process variance $\hat{\sigma}_k^2$ and the noise magnitude λ . Two methods are here used to find the optimal values of the hyperparameters, hereinafter referred to as “Full” and “Partial”. The optimization of the hyperparameters is performed by calling the NLOpt library (available online at <http://ab-initio.mit.edu/nlopt>) and implementing a sequential global–local approach: first, the search space is globally explored by means of the evolutionary strategy ESCH (da Silva Santos et al. 2010); then, starting from the best solution of the ESCH algorithm, a local refinement is carried out with a reviewed version of the Nelder-Mead simplex algorithm (Richardson and Kuester 1973).

Full Optimization

This formulation determines the regression parameters based on an optimality condition and fits all other covariance parameters (length scales, process variance and noise level) through maximization of the likelihood function. The likelihood formula for a Gaussian process with a regression mean function is given by:

$$\log p(f|x; \theta_p) = -\frac{1}{2} \mathbf{f}^T \mathbf{K}^{-1} (\mathbf{f} - \mathbf{H} \hat{\beta}) - \frac{1}{2} \log |\mathbf{K}| - \frac{1}{2} \log |\mathbf{A}| - \frac{M - S}{2} \log 2\pi$$

where M is the number of training points, S is the number of terms in the regression and the regression matrix A is defined as:

$$\mathbf{A} = \mathbf{H}^T \mathbf{K}^{-1} \mathbf{H}$$

The optimal regression parameters are given by:

$$\hat{\beta} = \mathbf{A}^{-1} \mathbf{H}^T \mathbf{K}^{-1} \mathbf{f}$$

Partial Optimization

This formulation determines the process variance and regression parameters based on the optimality condition and only performs optimization over the covariance length scales θ_p . The likelihood formula reduces to:

$$\log p(f|x; \theta_p) = -\frac{M}{2} \log \hat{\sigma}_k^2 - \frac{1}{2} \log |\hat{\mathbf{K}}| - \frac{M}{2} - \frac{M}{2} \log 2\pi$$

where the optimal process variance has been estimated as:

$$\hat{\sigma}_k^2 = \frac{(\mathbf{f} - \mathbf{H}\hat{\beta})^T \hat{\mathbf{K}}^{-1} (\mathbf{f} - \mathbf{H}\hat{\beta})}{M}$$

and

$$\mathbf{K} = \hat{\sigma}_k^2 \hat{\mathbf{K}}$$

This final formula is a function of the length scales, θ_p , and the noise level ratio λ . The optimization is performed only over the length scales and the noise level ratio is fixed throughout the optimization. A typical choice is to set the noise level λ to a small fraction of the process variance $\hat{\sigma}_k$.

Radial Basis Function Network

A Radial Basis Function is a real valued function whose value depends on the Euclidean distance from a point called centre. A RBF network uses a linear combination of radial functions. A RBF model can be expressed as

$$f(\mathbf{x}, \theta_1, \dots, \theta_M, \lambda) = \sum_{i=1}^M k_i(\lambda) r(|\mathbf{x} - \mathbf{x}_i|, \theta_i) \tag{6}$$

where the approximating function is represented by a sum of M RBFs r , each associated with a different center \mathbf{x}_i , weighted by real valued weights k_i (regularized through parameter λ) and characterized by width parameters θ_i . Hence, an RBF network can be defined as a weighted sum of translations of radially symmetric basis function. Typical RBFs kernel r used here are:

$$r(d, \theta) = \begin{cases} \exp\left(-\frac{d^2}{\theta^2}\right) & \text{Gaussian} \\ \sqrt{1 + \frac{d^2}{\theta^2}} & \text{Multi-quadric} \\ \frac{1}{\sqrt{1 + \frac{d^2}{\theta^2}}} & \text{Inverse multi-quadric} \\ \left(\frac{d}{\theta}\right)^2 \ln \frac{d}{\theta} & \text{Thin plate spline} \\ 1 - 30\left(\frac{d}{\theta}\right)^2 - 10\left(\frac{d}{\theta}\right)^3 + \\ + 45\left(\frac{d}{\theta}\right)^4 - 6\left(\frac{d}{\theta}\right)^5 - 60\left(\frac{d}{\theta}\right)^3 \log\left(\frac{d}{\theta}\right) & \text{Wendland } C^2 \text{ thin plate spline} \end{cases}$$

Once decided the RBF kernel and supposing that the “optimal” width parameters have been already computed in some way, the RBF network is defined only by the weights k_i . They are made function of a regularization parameter λ (also known as ridge regression parameters in the RBF literature) to avoid overfitting and improve the interpolation matrix conditioning. Indeed, the weights can be found by imposing the interpolation condition (Fasshauer and Zhang 2007) on the training set which in turn results in solving the linear system:

$$\mathbf{R}\mathbf{k} = \mathbf{f} \quad (7)$$

where

$$\mathbf{R} = \begin{Bmatrix} r(0, \theta_1) + \lambda & \dots & r(|\mathbf{x}_1 - \mathbf{x}_M|, \theta_M) \\ r(|\mathbf{x}_2 - \mathbf{x}_1|, \theta_1) & \dots & r(|\mathbf{x}_2 - \mathbf{x}_M|, \theta_M) \\ \vdots & \vdots & \vdots \\ r(|\mathbf{x}_M - \mathbf{x}_1|, \theta_1) & \dots & r(0, \theta_M) + \lambda \end{Bmatrix}$$

$\mathbf{k} = [k_1, k_2, \dots, k_M]^T$ are the RBF weights and $\mathbf{f} = [f_1, f_2, \dots, f_M]^T$ are the function values at the training points.

The width parameters have a significant influence both on the accuracy of the RBF model and on the conditioning of the solution matrix. In particular, it has been found (Gutmann 2001) that interpolation errors become high for very small and very large values of θ , while the condition number of the coefficient matrix increases with increasing values of θ . Therefore, they have to be “optimal” in the sense that a tuning of the width parameters is needed to find the right trade-off between interpolation errors and solution stability (Fasshauer and Zhang 2007). Generally speaking, two cases can be considered:

- identical scalar widths $\theta_i = \theta$ are used for all RBF kernels;
- different scalar width θ_i is used for each RBF kernel.

Here, the first option is chosen, therefore in the following a unique scalar width θ will be considered for each RBF center. An accurate RBF model is obtained by letting the algorithm autonomously choose the kernel function type *and* optimizing the width parameters. The algorithm is based on the Leave-One-Out cross-validation strategy to compute an error norm to be minimized; the procedure is similar to the one described in (Tenne and Armfield 2008) and is here outlined:

1. all the aforementioned kernel functions are used for training on the current training set;
2. the Leave-One-Out (LOO) error norm is considered as merit function to determine the best combination of RBF kernel and width parameter. The optimal RBF network is thus selected by choosing the width parameter which give the lowest LOO error norm, defined as:

$$\varepsilon_{LOO}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M, \theta, \lambda) = \sqrt{\frac{1}{M} \sum_{j=1}^M [f_j - \hat{f}_{-j}(\mathbf{x}_j, \theta, \lambda)]^2}$$

where f_j is the value of the function at the j^{th} training site \mathbf{x}_j and \hat{f}_{-j} is the RBF prediction at \mathbf{x}_j when the model is trained without \mathbf{x}_j and f_j . The computation of the M terms \hat{f}_{-j} does not require to train M RBF models, indeed it can be computed effortlessly thanks to Rippa’s formula (Rippa 1999);

3. for each kernel, the width parameter θ and regularization parameter λ are found by solving:

$$\min_{\theta, \lambda} \varepsilon_{LOO}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M, \theta, \lambda) \quad (8)$$

The optimization is performed by using the same algorithms for searching the Kriging hyperparameters.

POD + Radial Basis Function Networks

The Proper Orthogonal Decomposition (POD) is used to extract the main features of a set of computed flow fields as a series of POD basis vectors with associated coefficients (Iuliano 2011; Iuliano and Quagliarella 2011). Given the three spatial coordinates (ξ, ν, ζ) of the computational mesh points and the general snapshot vector \mathbf{s} , let $\{\mathbf{x}_j\}$ be a set of design vectors (e.g., sampled from the design space with a DoE technique) and $\{\mathbf{s}_j\}$ the corresponding snapshot, i.e. column vectors containing the volume grid and flow variables as obtained from a CFD solution:

$$\begin{aligned} \mathbf{s} &= (\mathbf{s}_{\text{grid}}, \mathbf{s}_{\text{flow}})^T \\ \mathbf{s}_{\text{grid}} &= (\xi_1, \dots, \xi_q, \nu_1, \dots, \nu_q, \zeta_1, \dots, \zeta_q) \\ \mathbf{s}_{\text{flow}} &= (\rho_1, \dots, \rho_q, \rho\xi'_1, \dots, \rho\xi'_q, \rho\nu'_1, \dots, \rho\nu'_q, \\ &\quad \rho\zeta'_1, \dots, \rho\zeta'_q, p_1, \dots, p_q) \end{aligned}$$

where q is the number of mesh nodes involved in the POD computation, ρ is the flow density, (ξ', ν', ζ') are the three Cartesian velocity components and p is the static pressure. The computational mesh has been included in the POD snapshot to let the SVD basis catch the coupling effects between space location and state field. Hence, once the surrogate model is built, not only a flow field can be computed, but also an approximation of the volume mesh. Such a surrogate model would be able to catch, although in a reduced order form, the cross effects of geometry modification and aerodynamic flow change. As the total number of variables is eight (three mesh variables and five flow variables), the global size of the snapshot is $N = 8 \times q$.

Starting from the vectors $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_M$ obtained by CFD expensive computations for a representative set of design sites $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M$, finding a Proper Orthogonal Decomposition means to compute a linear basis of vectors to express any other $\mathbf{s}_j \in \mathbb{R}^N$ with the condition that this basis is optimal in some sense. To compute the optimal basis, we first define the snapshot deviation matrix

$$\mathbf{P} = (\mathbf{s}_1 - \bar{\mathbf{s}} \quad \mathbf{s}_2 - \bar{\mathbf{s}} \quad \dots \quad \mathbf{s}_M - \bar{\mathbf{s}})$$

where the ensemble mean vector is computed as

$$\bar{\mathbf{s}} = \frac{1}{M} \sum_{j=1}^M \mathbf{s}_j$$

The POD decomposition is obtained by taking the singular value decomposition (SVD) of \mathbf{P}

$$\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U} \begin{pmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_M \\ 0 & \cdots & 0 \end{pmatrix} \mathbf{V}^T \quad (9)$$

with $\mathbf{U} \in \mathbb{R}^{N \times N}$, $\mathbf{V} \in \mathbb{R}^{M \times M}$, $\mathbf{\Sigma} \in \mathbb{R}^{N \times M}$ and the singular values $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_M \geq 0$. The POD basis vectors, also called POD modes, are the first M column vectors of the matrix \mathbf{U} , while the POD coefficients $\alpha_i(\mathbf{x}_j)$ are obtained by projecting the snapshots onto the POD modes:

$$\alpha_i(\mathbf{x}_j) = (\mathbf{s}_j - \bar{\mathbf{s}}, \boldsymbol{\phi}_i) \quad (10)$$

If a fluid dynamics problem is approximated with a suitable number of snapshots from which a rich set of basis vectors is available, the singular values become small rapidly and a small number of basis vectors are adequate to reconstruct and approximate the snapshots as they preserve the most significant ensemble energy contribution. In this way, POD provides an efficient mean of capturing the dominant features of a multi-degree of freedom system and representing it to the desired precision by using the relevant set of modes. The reduced order model is derived by projecting the CFD model onto a reduced space spanned by only some of the proper orthogonal modes or POD eigenfunctions. This process realizes a kind of lossy data compression through the following approximation

$$\mathbf{s}_j \simeq \bar{\mathbf{s}} + \sum_{i=1}^{\hat{M}} \alpha_i(\mathbf{x}_j) \boldsymbol{\phi}_i \quad (11)$$

where

$$\hat{M} \leq M \implies \sum_{i=1}^{\hat{M}} \sigma_i^2 \geq \varepsilon \sum_{i=1}^M \sigma_i^2 \quad (12)$$

and ε is a predefined energy level. In fact, the truncated singular values fulfils the relation

$$\sum_{i=\hat{M}+1}^M \sigma_i^2 = \varepsilon \hat{M}$$

If the energy threshold is high, say over 99% of the total energy, then \hat{M} modes are adequate to capture the principal features and approximately reconstruct the dataset. Thus, a reduced subspace is formed which is only spanned by \hat{M} modes.

Equation 11 allows to get a POD approximation of any snapshot s_j belonging to the ensemble set. Indeed, the model does not provide an approximation of the state vector

at design sites which are not included in the original training dataset. In other words, the POD model by itself does not have a global predictive feature, i.e. over the whole design space. As the aim is to exactly reproduce the sample data used for training and to consistently catch the local data trends, a Radial Basis Function (RBF) network answers to these criteria and has been chosen as POD coefficients interpolation. The procedure to build optimal RBF models for POD modal coefficients is the same as described in Section “[Radial Basis Function Network](#)”.

As a results, the pseudo–continuous prediction of the flow field at a generic design site \mathbf{x} is then expressed as:

$$\mathbf{s}(\mathbf{x}) = \bar{\mathbf{s}} + \sum_{i=1}^{\hat{M}} \alpha_i(\mathbf{x}) \phi_i \quad (13)$$

This provides an accurate surrogate model which combines design of experiments for sampling, CFD for training, POD for model reduction and RBF network for global approximation. In conclusion, an explicit, global, low–order and physics–based model linking the design vector and the state vector has been derived and will be used as surrogate model. Examples of application and validation of the proposed POD/RBF surrogate model have been already provided in recent papers (Iuliano 2011; Iuliano and Quagliarella 2013).

Adaptive Sampling Strategy

Supposing that a surrogate model has been already trained, the training set is enriched by adding new samples, then the surrogate model is rebuilt and globally optimized. Hence, an iterative scheme is used for surrogate-based optimization: in the previous iteration, optimal candidates from the surrogate minimization are selected and passed to the next iteration; in the next iteration, the new samples are evaluated via the true, high-fidelity model and re-injected into the training set upon which the surrogate is updated. The aim of such an iterative scheme is to increase the quality and potential of the surrogates to be minimized, presumably driving to true optimality quickly. Of course, as this approach relies totally on the surrogate model and its prediction, it may drive the process towards local minima from which the surrogate model can no longer escape.

The weak point is considering the enrichment with new samples as a purely “exploitation” process and ignoring the “explorative” behaviour. Prior or during the optimization on the surrogate, we need to mix the knowledge from the available data, the surrogate prediction and an estimation of its predictive capability: we need to have a “smarter” selection of new points. However, the strategy for updating a surrogate model is heavily dependent on its type and scope and, in principle, has to be tailored on it. Indeed, the addition of new samples must follow some specific criteria that may be very different depending on the purpose of the training process.

For instance, Latin Hypercube Sampling has been designed to satisfy space-filling requirements and obtain a good coverage of the design space.

The present approach gives emphasis to the optimization process by proposing sampling strategies which are able to “adapt” to the response function. Most of the adaptive sampling approaches pursue the exploration/exploitation trade-off, where exploration means sampling away from available data, where the prediction error is supposedly high, while exploitation means trusting the model prediction, thus sampling where the surrogate provides global minima. It is clear that a trade-off between the two behaviors is needed: indeed, exploration is useful for global searching, but it may lead to unveil uninteresting regions of the design space; on the other hand, exploitation helps to improve the local accuracy around the predicted optima, but it may result in local minima entrapment.

Here, balanced explorative in-fill criteria are designed for a generic surrogate model and are formulated in terms of an auxiliary function which has to be maximized. The balanced criterion, hereinafter referred to as “EI-like”, has been designed to mimic the same rationale of the Expected Improvement criterion, usually coupled to a Kriging-based surrogate in the well-known EGO algorithm by Jones (1998). The present approach, represents a generalization of that method as, for a generic surrogate model, the information about the uncertainty of the surrogate is not available, while a Kriging model, being a Gaussian process, provides an estimate of the prediction variance together with the prediction itself. The auxiliary function, also referred to as potential of improvement, is designed to have the same form of the Expected Improvement function.

Given \mathbf{x} the generic design space location, $\hat{f}(\mathbf{x})$ the surrogate response, X_n the dataset of the training samples collected so far, F_{X_n} the corresponding values of the true objective function, f_{max} and f_{min} the maximum and minimum values in F_{X_n} , the potential of improvement function (“EI-like” function) is defined as follows:

$$v(\mathbf{x}, \hat{f}(\mathbf{x}), X_n, F_{X_n}) = [f_{min} - \hat{f}(\mathbf{x})] \Phi \left[\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})} \right] + \hat{s}(\mathbf{x}) \phi \left[\frac{f_{min} - \hat{f}(\mathbf{x})}{\hat{s}(\mathbf{x})} \right]$$

where $\hat{s}(\mathbf{x})$ is an estimate of the prediction error and $\Phi(\mathbf{x})$ and $\phi(x)$ are respectively the cumulative distribution and probability density functions of a standard normal distribution. The prediction error is estimated as follows:

$$\hat{s}(\mathbf{x}) = L(\mathbf{x}) \frac{\min_{\mathbf{x}_i \in X_n} \|\mathbf{x} - \mathbf{x}_i\|_2}{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} \|\mathbf{x}_i - \mathbf{x}_j\|_2} \exp \left(-\gamma \frac{\max_{\mathbf{x}_i, \mathbf{x}_j \in X_n} \|\mathbf{x}_i - \mathbf{x}_j\|_2}{\min_{\mathbf{x}_i \in X_n} \|\mathbf{x} - \mathbf{x}_i\|_2} \right)$$

where $L(\mathbf{x})$ is an estimate of the Lipschitz constant at \mathbf{x} and γ is a tuning parameter. The Lipschitz constant is defined as:

Definition 1 Given a domain D and a function f defined in D , the Lipschitz constant is the smallest constant $L > 0$ in the Lipschitz condition, namely the non negative number:

$$L_{f,D} := \sup_{\substack{x_1, x_2 \in D \\ x_1 \neq x_2}} \frac{|f(x_1) - f(x_2)|}{|x_1 - x_2|}$$

The following algorithm has been designed to obtain an estimate of the Lipschitz constant at each training sample:

Algorithm 1 Lipschitz constant estimation

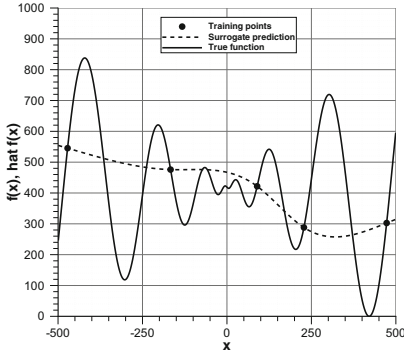
- 1: compute the K-means clusters $K_{j,j=1,r}$ of the set $X_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ with $r = \text{int}(\frac{n}{q})$
 - 2: **for all** sample $\mathbf{x}_i \in X_n$ **do**
 - 3: Say K_i the cluster containing \mathbf{x}_i
 - 4: **for all** sample $\mathbf{x}_j \in K_i, \mathbf{x}_j \neq \mathbf{x}_i$ **do**
 - 5: compute $L_{ij} = \frac{|f(\mathbf{x}_i) - f(\mathbf{x}_j)|}{|\mathbf{x}_i - \mathbf{x}_j|}$
 - 6: **end for**
 - 7: Set $L(\mathbf{x}_i) = \max_j L_{ij}$
 - 8: **end for**
-

Finally, in order to extend the estimation to a generic location \mathbf{x} , it is assumed that $L(\mathbf{x}) = L(\mathbf{x}_{nn})$ where $\mathbf{x}_{nn} = \text{argmin}_{\mathbf{x}_i \in X_n} |\mathbf{x}_i - \mathbf{x}|$. The function $\hat{s}(\mathbf{x})$ mimics the Gaussian Process prediction error and has been designed to quickly increase with increasing distance from an available sample. Moreover, its order of magnitude is comparable to the actual values of the objective function. The adaptive in-fill process is organized as follows: a huge Latin Hypercube Sampling dataset (e.g., 500 times the dimension of the design space) is obtained and the values of the potential of improvement is computed at each point (this requires limited computational effort as the auxiliary function only depends on the surrogate prediction, which is fast to obtain, and on the true objective function values at already collected points); hence, the new sample is located where the maximum value of the auxiliary function is met:

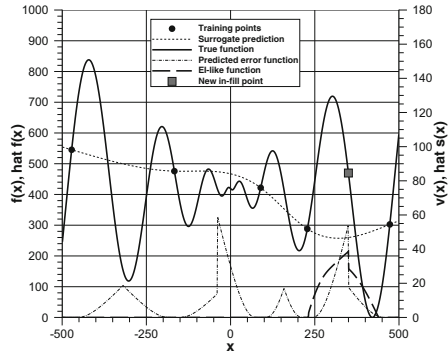
$$\mathbf{x}_{n+1} = \underset{\mathbf{x}}{\text{argmax}} \nu(\mathbf{x}, \hat{f}(\mathbf{x}), X_n, F_{X_n})$$

In order to avoid the duplication of the updating samples when iterating the in-fill process, the seed of the Latin Hypercube is changed at each iteration.

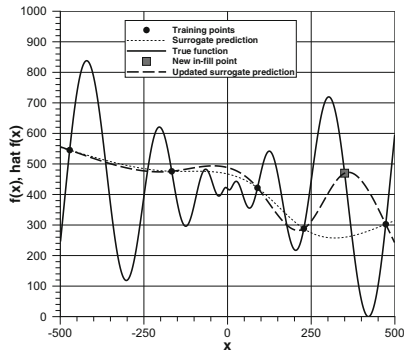
Figure 1 provides an example of surrogate updating by maximization of the EI-like criterion. The one-dimensional Schwefel function is used as test function with 5 initial training points. The trained surrogate (here, a Kriging model) does not capture the local non-linear features of the true function, but a certain trend to predict low values where the true optimum resides is observed (Fig. 1a). The Lipschitz-based prediction error function and the EI-like function are reported in Fig. 1b: by taking the maximum of the EI-like function, a new in-fill point (grey square) is obtained and the surrogate is updated (Fig. 1c). This first iteration seem to not improve the prediction too much: in fact, it provides information about the high non-linearity of the true function around the optimum as the surrogate model now “knows” that the function is rapidly changing in that region. After 10 iterations of the in-fill process,



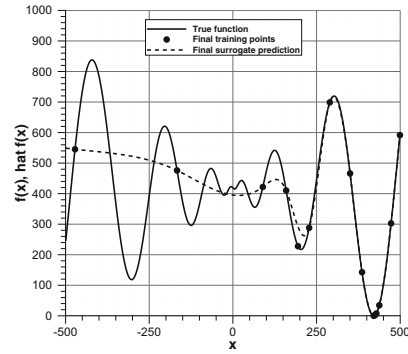
(a) True and surrogate functions $f(\mathbf{x})$, $\hat{f}(\mathbf{x})$ and training dataset $\{X_n, F_{X_n}\}$



(b) Functions $\hat{s}(\mathbf{x})$ and $v(\mathbf{x}, \hat{f}(\mathbf{x}), X_n, F_{X_n})$



(c) New in-fill point and updated surrogate



(d) Updated surrogate with 10 in-fill points

Fig. 1 Example of surrogate updating by maximization of the Lipschitz in-fill criterion on the 1D Schwefel function

the true optimum is perfectly captured as well as the whole trend of the function past $x = 250$ (Fig. 1d).

Surrogate-Based Optimization

The workflow of the surrogate-based shape optimization (SBSO) is depicted in Fig. 2. The method is centered on the surrogate training database which is continuously fed and updated throughout the search and optimization process. As a first step, it is initialized with a space-filling design of experiment (e.g., a Latin Hypercube Sampling or a Latinized Central Voronoi Tessellation): typically, according to literature results and authors past experience, the number of initial samples (n_{apr}) should not exceed

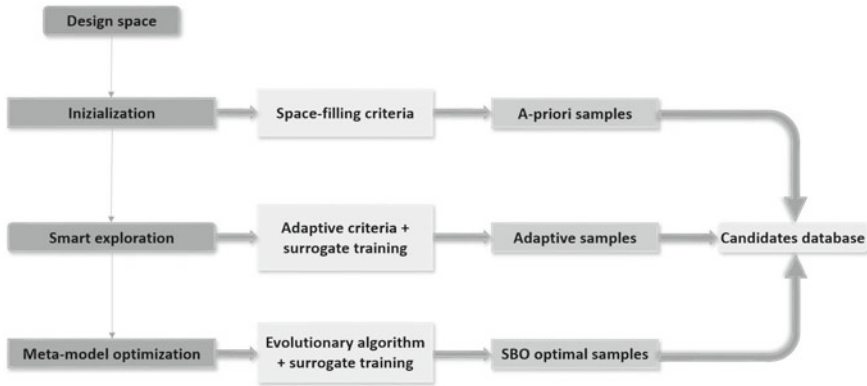


Fig. 2 Workflow of surrogate-assisted optimization

one-third of the total computational budget. The evaluation of the response function corresponding to a given sample is made as follows:

- a geometry parameterization module (CST approach, Kulfan 2008) transforms the design vector (i.e., the training sample) into the actual component shape;
- a batch scripting procedure is launched within ANSYS ICEM CFD package to generate the CAD surface and the volume mesh with fixed sizes and topology;
- a CFD computation is launched with the in-house ZEN CFD flow solver (Catalano and Amato 2003);
- once the simulation has converged, the objective function (usually depending on computed aerodynamic coefficients) and the flow field snapshots are collected according to the specification of the design problem.

As multiple training samples have to be evaluated simultaneously, the process can be executed in parallel to speed up the simulation. Once the evaluation process has finished, the selected surrogate model can be built as described in Section “Surrogate Models”.

The workflow in Fig. 2 embeds two internal cycles, namely the adaptive sampling and the optimization update. These iterative phases reflect two different needs: first, providing an improved and reliable model to the optimizer; then, iterating the optimizer to refine the optimum search. The first cycle consists of updating the design solutions database by applying in-fill criteria (as described in Section “Adaptive Sampling Strategy”) and providing n_{adpt} new design candidates. The condition to exit from this internal loop is based either on predefined levels of improvement or on computational budget considerations.

The second cycle (database updating by optimization) allows for including n_{opt} sub-optimal samples suggested by sequentially optimizing the meta-model and re-injecting the best candidate in the training database: this phase should lead to the final exploitation of the design space region where the “true” optimum resides. The loop terminates either when the residual of the objective function of the predicted

optima falls below a predefined threshold or when the computational budget limit has been reached. The total computational budget n_{tot} is fixed a-priori and is equal to $n_{tot} = n_{apr} + n_{adpt} + n_{opt}$.

The optimizer consists of an hybrid algorithm implemented within the in-house library ADGLIB (Quagliarella et al. 2004): a genetic algorithm is used for global search and the CMA-ES (Hansen 2006) algorithm acts as a local search operator. During the evaluation of the population, the CMA-ES algorithm is triggered with a predefined activation probability to improve the current best solution.

Numerical Results

The public domain 3rd Drag Prediction Workshop DPW-W1 wing (Epstein et al. 2008) has been selected as the initial geometry for aerodynamic optimization. Reference data for this wing are shown in Table 1. The nominal flow conditions are prescribed at two design points:

1. Mach = 0.76, Reynolds = 5×10^6 , $C_{L,0,1} = 0.5$, $C_{D,0,1} = 0.0241$, $C_{M,0,1} = -0.07$
2. Mach = 0.78, Reynolds = 5×10^6 , $C_{L,0,2} = 0.5$, $C_{D,0,2} = 0.0279$, $C_{M,0,2} = -0.08$

where $C_{L,0,k}$, $C_{D,0,k}$, $C_{M,0,k}$ are the lift, drag and pitching moment coefficient of the baseline wing at the k -th design point. The objective function to be minimized is:

$$f(\mathbf{x}) = \sum_{k=1}^2 \frac{1}{2} \frac{C_{D,k} + C_{DM,k} + C_{DL,k}}{C_{L,k}} \frac{C_{L,0,k}}{C_{D,0,k}} \quad (14)$$

$$C_{DM,k} = 0.01 \max(0, C_{M,0,k} - C_{M,k})$$

$$C_{DL,k} = 0.1 \max(0, C_{L,0,k}^2 - C_{L,k}^2)$$

Geometric constraints are also implemented in terms of minimum value of the wing section maximum thickness (=13.5%) and of the beam thickness constraints at two locations along the wing airfoil chord (=12% thickness ratio at 20% wing section chord and 5.9% thickness ratio at 75% wing section chord).

Table 1 Reference data for DPW wing

Wing area	290,322 mm ²
Mean aerodynamic chord	197.55 mm
X_{ref} for moments	154.24 mm (from root l.e.)
Semi-span length	762 mm
Aspect ratio	8.0

Geometry Parameterization and Mesh Generation

The CST approach allows to identify and isolate the general features which similar shapes have in common (e.g., round/sharp nose, cross section area distribution) and separate the contribution introduced by the real shape change. This “factorisation” is carried out through the definition of a “class” function and a “shape” function, whose product give then the real shape. More details can be found in (Kulfan 2008). In the present case, the wing shape is described by 36 shape variables + 1 variable to modify the twist angle at the wing tip. In order to build the wing shape, three locations along the non-dimensional span length η are selected ($\eta = 0.0, 0.5, 1.0$) and, once given the design weights, the sectional shapes at those three sections are extracted from the analytical CST representation as a set of points; hence, the points are read in ANSYS ICEM CFD and a sequence of parametric commands are executed through a batch script to generate and export the computational mesh. The volume mesh is made of 8 blocks, a family of two grids is defined: the coarse and fine mesh consist respectively of 712,448 cells and 2,959,872 cells. A sketch of the surface mesh distribution is shown in Fig. 3. Both meshes are conceived to respect the $y^+ = O(1)$ condition, as also shown in the figure where the contour map of y^+ distribution on the wing surface is depicted. The coarse mesh will be used for optimization studies, while the fine mesh will provide more accurate comparisons of the aerodynamic flow for optimized shapes at the end of the optimization process.

Optimization Results

Four different surrogate-based simulations have been carried out and detailed in Table 2. The standard EGO algorithm has been included to set a reference level. The total computational budget is fixed at 500 CFD calls as well as the size of the initial training set is common to all methods and equal to 216. This will allow a fair comparison between the single method capabilities to search the design space with equivalent computational effort. When the present surrogate-based optimization method is used, the EI-like in-fill criterion is adopted for testing purposes.

Figure 4 shows the optimization histories in terms of the progression of the minimum objective function value found in the training database along the iterations. The unit value represents the level of the baseline DPW wing shape. The models have roughly the same pattern, with the POD/RBFN model slightly outperforming the others. The different approach between EGO and the present method is clearly observed: EGO pushes to minimize the objective function from the beginning of the updating phase (i.e., after the initial 216 samples evaluation) having a steady and continuous improvement; on the other hand, the present surrogate-based method achieves a significant contribution to the descent in the final 100 samples, where optimization search is actively working, leaving to the intermediate 184 (adaptive)

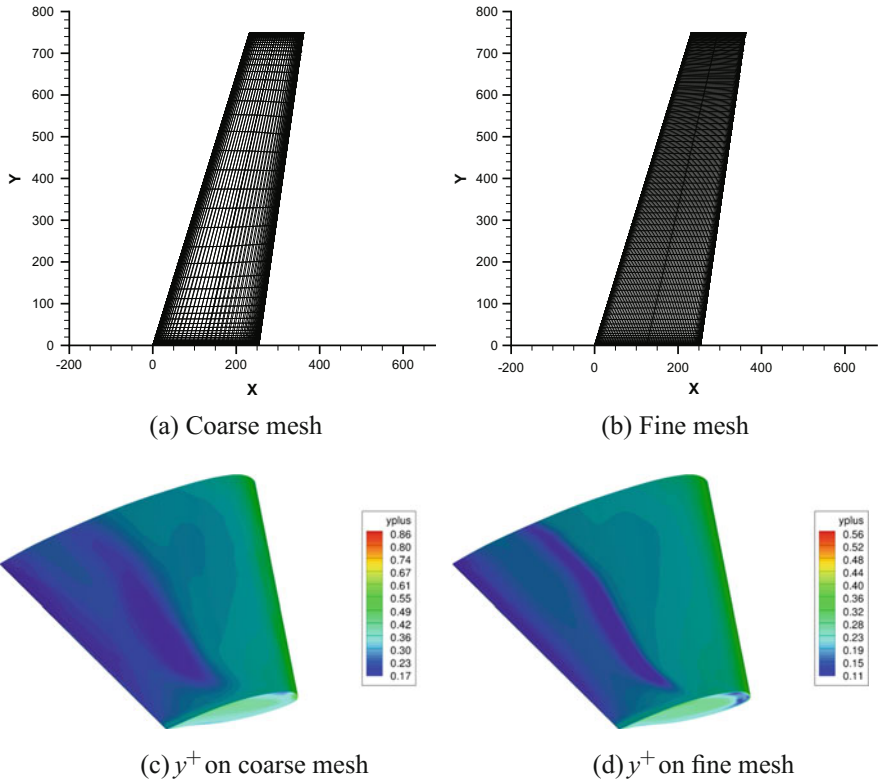


Fig. 3 Surface mesh and y^+ distribution on DPW wing surface

Table 2 Optimization setup

Method	Surrogate	In-fill criteria	n_{apr}	n_{adpt}	n_{opt}	Total CFD calls
EGO	Kriging	EI	216	–	284	500
Present SBO	Kriging	EI-like	216	184	100	500
Present SBO	RBFN	EI-like	216	184	100	500
Present SBO	POD/RBFN	EI-like	216	184	100	500

samples the freedom to improve the surrogate quality. At the end of the process, each of the three present SBO methods reaches better results than EGO.

Table 3 propose a comparison of the aerodynamic coefficients and objective function value for all optimal candidates. The keypoint of the optimization task is the drag reduction on DP2, where the improvement is much larger. Slight differences are noticed on pitching moment coefficients as no optimum satisfies the constraint. Indeed, in the minimization problem formulation (Eq. 14), the pitching moment

Fig. 4 Convergence histories of surrogate-based optimizations

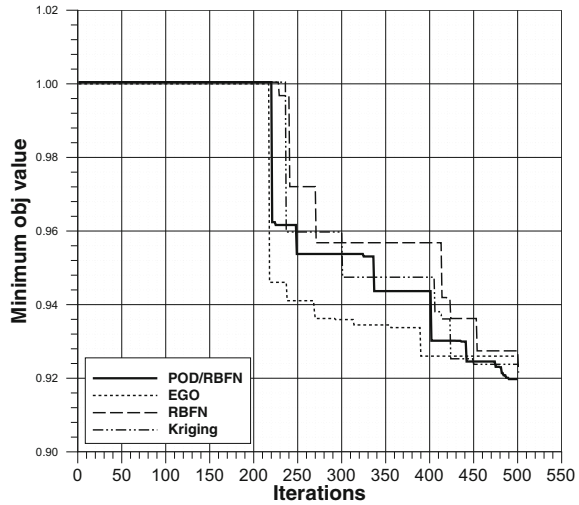


Table 3 Aerodynamic performances of optimal candidates

Design	DP1			DP2			Obj. value
	$C_{L,1}$	$C_{D,1}$	$C_{M,1}$	$C_{L,2}$	$C_{D,2}$	$C_{M,2}$	
Baseline	0.500	0.0241	-0.0813	0.500	0.0279	-0.0880	1.0
EGO opt.	0.500	0.0231	-0.0942	0.500	0.0244	-0.099	0.926
RBFN opt.	0.500	0.0231	-0.102	0.500	0.0241	-0.108	0.921
Kriging opt.	0.500	0.0232	-0.095	0.500	0.0242	-0.100	0.923
POD/RBFN opt.	0.500	0.0231	-0.086	0.500	0.0243	-0.0918	0.920

constraint has been implemented as a soft penalty (1 drag counts penalty for 0.01 variation in C_M), hence the method allows to exceed it if the gain in aerodynamic drag is more significant. Anyway, the most interesting result is that all optimal candidates show very similar performances: the relative difference in aerodynamic drag is within 1 count at DP1 and 3 counts at DP3.

Pressure contour maps for selected optimal candidates are depicted in Fig. 5. The inboard wing loading is slightly reduced on design point 1 and a significant decrease of the shock wave intensity is observed on the mid-outboard wing. By comparing the optimal solutions, it is quite evident that EGO and Kriging-based optima are indeed similar as the optimization relied on similar surrogates, even if the adaptive criterion for adding new samples is different. The POD/RBFN model is able to perform slightly better because it is a physics-based approach, i.e. it is fed not only with values of the objective function but mainly with computed flow fields. This peculiar aspect allows to inherit more information related to the nature of the governing equations

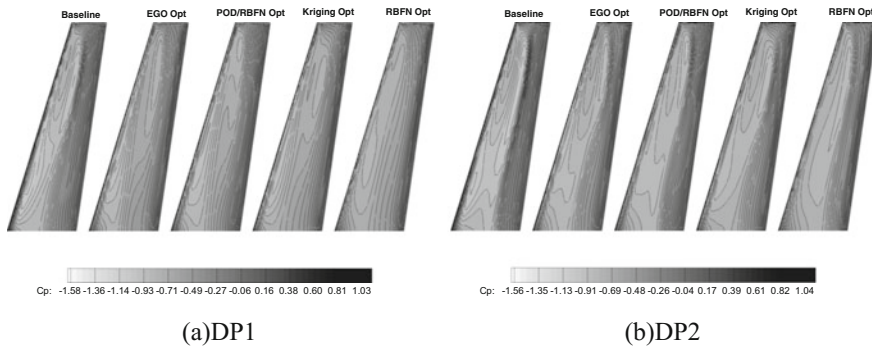


Fig. 5 Pressure coefficient contour maps

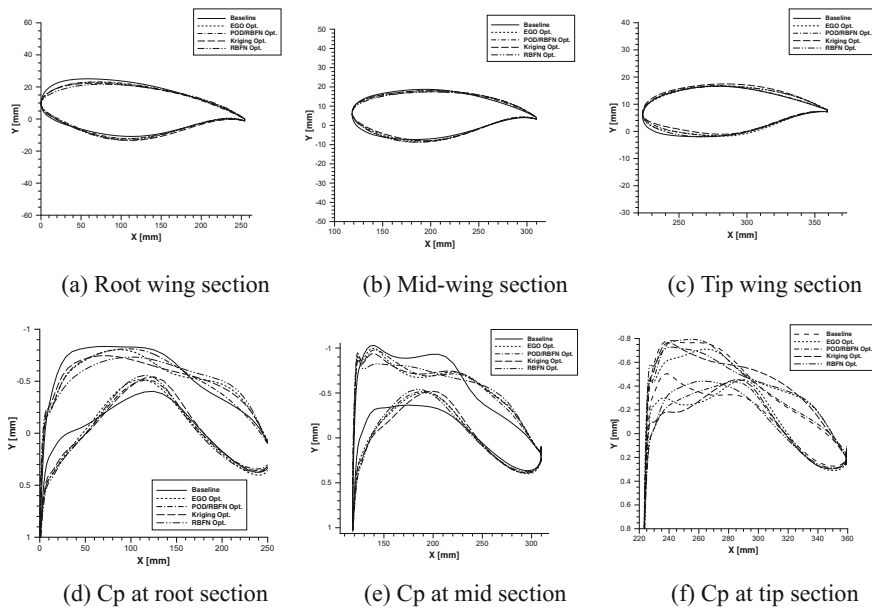


Fig. 6 Sectional airfoil geometry and Cp distribution

(e.g., flow field structure, shock-wave pattern, boundary layer characteristics) when reconstructing and predicting new solutions. Finally, Fig. 6 proposes a comparison of the local geometry and pressure coefficient solution of the optimal candidates. Three y-constant wing sections are selected, namely at wing root, mid-wing and tip locations. In terms of geometry modifications with respect to the baseline shape, an important reduction of the leading edge curvature is observable and a slight increase of the rear airfoil curvature near the wing tip (probably to recover the lift constraint). The twist angle at the wing tip has also been reduced for wing loading compensation.

Conclusions

The paper proposed a surrogate-assisted methodology suitable to aerodynamic shape optimization. Two scalar-valued surrogates (Kriging, RBFN) and a physics-based meta-model coupling Proper Orthogonal Decomposition and Radial Basis Functions interpolation have been used to predict approximate values of the objective functions throughout the optimization process. The training process has been conceived in three stages, namely a space-filling stage to initialize the surrogate, an adaptive sampling stage in which the model is gradually improved and a final iterative optimization stage where a sequence of improved surrogates are optimized. In the adaptive sampling phase, an in-fill criterion is designed to mimic the Expected Improvement Function maximization by re-formulating the surrogate prediction variance through the estimation of the Lipschitz constant.

An aerodynamic case has been proposed to test the methodology, consisting in the shape optimization of an isolated wing from the AIAA CFD Drag Prediction Workshops with 37 design variables and multi-point conditions. Despite the large scale and the complexity of the case, results are fully satisfactory because of either the obtained improvement (up to 10% on DP2) and the very limited computational cost (only 500 CFD calls).

Such results support the conclusion that surrogate models alone may not provide the right answer within an aerodynamic shape optimization context, especially if transonic viscous flow is considered. However, when coupled to smart adaptive sampling techniques, they allow to catch the basic trends of the objective function without penalizing the design space exploration: indeed, in complex design cases with high non-linearities and multi-modal landscapes, the latter has to be carefully balanced as it may result in unveiling promising regions as well as lead the optimizer to waste time in searching poor solutions.

References

- Booker AJ, Dennis JE, Frank PD, Serafini DB, Torczon V, Trosset MW (1999) A rigorous framework for optimization of expensive functions by surrogates. *Struct Multi Optim* 17:1–13. <https://doi.org/10.1007/BF01197708>
- Braconnier T, Ferrier M, Jouhaud J-C, Montagnac M, Sagaut P (2011) Towards an adaptive POD/SVD surrogate model for aeronautic design. *Comput Fluids* 40(1):195–209
- Catalano P, Amato M (2003) An evaluation of RANS turbulence modelling for aerodynamic applications. *Aerosp Sci Technol* 7:493–509
- da Silva Santos CH, Goncalves MS, Hernandez-Figueroa HE (2010) Designing novel photonic devices by bio-inspired computing. *IEEE Photonics Technol Lett* 22(15):1177–1179
- Epstein B, Jameson A, Peigin S, Roman D, Vassberg J, Harrison N (January, 2008) Comparative study of 3D wing drag minimization by different optimization techniques. In: 46th AIAA Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics and Astronautics
- Fasshauer GE, Zhang JG (2007) On choosing "optimal" shape parameters for RBF approximation. *Numer Algorithms* 45(1):345–368

- Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1–3):50–79
- Gutmann HM (2001) A radial basis function method for global optimization. *J Global Optim* 19:201–227
- Hansen N (2006) The CMA evolution strategy: a comparing review. In: Lozano J, Larranaga P, Inza I, Bengoetxea E (eds) *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, Springer, Berlin, pp 75–102
- Iuliano E (2011) Towards a POD-based surrogate model for CFD optimization. In: *Proceedings of the ECCOMAS CFD & optimization conference*. Antalya, Turkey
- Iuliano E, Quagliarella D (2011) Surrogate-based aerodynamic optimization via a zonal pod model. In: *Proceedings of the EUROGEN 2011 Conference*. Capua, Italy
- Iuliano E, Quagliarella D (2013) Proper orthogonal decomposition, surrogate modelling and evolutionary optimization in aerodynamic design. *Comput Fluids* 84:327–350
- Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Global Optim* 13:455–492
- Kulfan BM (2008) Universal parametric geometry representation method. *J Aircr* 45(1):142–158
- Mack Y, Goel T, Shyy W, Haftka R (2007) Surrogate model-based optimization framework: a case study in aerospace design. Springer, Berlin, pp 323–342
- Martin JD, Simpson TW (2005) Use of kriging models to approximate deterministic computer models. *AIAA J* 43(4):853–863
- Quagliarella D, Iannelli P, Vitagliano PL, Chinnici G (2004) Aerodynamic shape design using hybrid evolutionary computation and fitness approximation. In: *AIAA 1st intelligent systems technical conference*. American Institute of Aeronautics and Astronautics (AIAA), Chicago, IL (AIAA Paper 2004-6514)
- Richardson JA, Kuester JL (1973) Algorithm 454: the complex method for constrained optimization [e4]. *Commun. ACM* 16(8):487–489
- Rippa S (1999) An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv Comput Math* 11(2):193–210
- Robinson T, Willcox K, Eldred M, Haimes R (September, 2006) Multifidelity optimization for variable-complexity design. In: *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*. American Institute of Aeronautics and Astronautics
- Tenne Y, Armfield SW (2008) A versatile surrogate-assisted memetic algorithm for optimization of computationally expensive functions and its engineering applications. Springer, Berlin, pp 43–72
- Viana FAC, Haftka RT (2012) Watson LT Efficient global optimization algorithm assisted by multiple surrogate techniques. *J Global Optim* 56(2):669–689

Study of the Influence of the Initial a Priori Training Dataset Size in the Efficiency and Convergence of Surrogate-Based Evolutionary Optimization



Daniel González-Juarez and Esther Andrés-Pérez

Abstract The development of an automatic geometry optimization tool for efficient aerodynamic shape design, supported by Computational Fluid Dynamic (CFD) methods is nowadays an attractive research field, as can be observed from the increasing number of scientific publications during the last years. Surrogate-based global optimization methods have demonstrated a huge potential to reduce the actual number of CFD runs, and therefore drastically speed-up the design process. Nevertheless, surrogates need initial high fidelity data sets to be built and to reach a proper accuracy. This work presents a study on the influence of the initial training dataset size in the proposed approach behavior. This approach is based on the use of Support Vector Machines (SVMs) as the surrogate model for estimating the objective function, in combination with an Evolutionary Algorithm (EA) and an adaptive sampling technique focused on optimization called the Intelligent Estimation Search with Sequential Learning (IES-SL). Several number of training points have been fixed to check the convergence, the accuracy and the objective function reached by the method.

Introduction

Aerodynamic shape optimization by means of automatic tools is an industrial relevant field that has to breast several challenges. Some of these challenges are: how to handle deformations in certain regions (such as intersections between wing and fuselage or pylon/nacelle), how to reduce the number of CFD runs required for performing aerodynamic design optimization or how to tackle integrated components. Furthermore, surrogate-based optimization methods require several barriers to be broken when applied to complex configurations, such as the called

D. González-Juarez (✉)

Fluid Dynamic Brach, National Institute for Aerospace Technology (INTA), Madrid, Spain
e-mail: gonzalezjd@inta.es

E. Andrés-Pérez

Engineering Department, Ingeniería de Sistemas para la Defensa de España S.A. (ISDEFE-INTA), Universidad Politécnica de Madrid UPM, Madrid, Spain
e-mail: eandres@isdefe.es

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_12

“curse of dimensionality”, the ability of surrogates to handle a high number of design parameters, efficient constraints handling (Parr et al. 2010), and the proper exploration and exploitation of the whole design space.

In the case of surrogate-based optimization (SBO) methods, the surrogate prediction is also highly influenced by training set size. A huge training set with a proper design space distribution ensures reaching a global optimum, but requires a vast computational cost to be built. On the other hand, a small training set is fast to be built but the accuracy is not enough for optimization purposes. A solution to this issue must be found for the suitable implementation of this method in the aeronautical industry.

In this work, Support Vector Machines (SVM) combined with Evolutionary Algorithms (EAs) and an adaptive sampling method, called Intelligent Estimation Search with Sequential Learning (IES-SL), is proposed. The approach is applied to the multipoint optimization of one typical test case, i.e., the transonic RAE 2822 airfoil. The aim of this work is to provide an analysis of the training set size influence in the behavior of the IES-SL approach proposed.

This paper is structured as follows. In Section “[Literature Review](#)”, a review of the recent research efforts in SBO applied to aircraft design is presented. Section “[Surrogate-Based Optimization Strategy](#)” presents the applied SBO strategy and Section “[Numerical Results](#)” collects the study results. Finally, the conclusions extracted from the results are summarized in Section “[Conclusions](#)”.

Literature Review

Recent Research Efforts in SBO Applied to Aircraft Design

Some recent efforts in SBO for aerodynamic shape design includes, e.g., a physics-based surrogates applied to the drag minimization of NACA 0012 and RAE 2822 airfoils in transonic flow conditions (Leifsson et al. 2014). In this work, the geometries were parameterized using PARSEC involving 5–10 design parameters. SBO strategies were applied for the drag minimization of the NLF0416 airfoil using 10 design variables (Li et al. 2001). Variable-fidelity computational fluid dynamics (CFD) combined with shape optimization strategy was applied to the optimization of a transonic airfoil parameterized by the NACA 4-digit definition with three design variables (Koziel and Leifsson 2013).

A surrogate based on proper orthogonal decomposition (POD) applied to the aerodynamic shape optimization of an airfoil is presented by Iuliano (Iuliano and Quagliarella 2013). The geometry was parameterized with 16 design variables defined with the CST method. An approach based on a combination of a genetic algorithm and an artificial neural network is presented by Jahangirian and Shahrokhi (2011). This approach was applied to the shape optimization of an airfoil, which was parameterized by a modified PARSEC involving 10 design variables.

Most of the SBO applications in aerodynamic shape optimization involve two-dimensional configurations, where the number of design variables is usually limited. Nevertheless, some applications to three-dimensional configurations can be found in literature. An investigation about SBO applied to a wing parameterized with 11 design variables was undertaken by Keane (2003). A multi-fidelity surrogate model applied to a three-dimensional wing optimization was addressed by Likeng and Zhenghong (2012). In this case, the design parameters were a combination of 12 variables using the CST method for three wing sections (root, hink and wing tip). Lukaczyk et al. (2014), proposed a method based on an active subspace for effectively searching the whole design space. The method is applied to the optimization of the ONERA M6 transonic wing, which was parameterized with 50 FFD design variables. The aim was to discover a low-dimensional linear subspace of the input space that explained the majority of the variability in the drag and lift coefficients. An SBO application to the aerodynamic shape design of a wing parameterized with volumetric non-uniform rational B-splines (NURBS) was presented by current authors (Andrés-Pérez and Iuliano 2015). Also, in (González-Juárez et al. 2015; Andrés-Pérez et al. 2016) current authors present an application study about the influence of number and location of the design parameters in the behaviour of the IES-SL method applied to the aerodynamic shape optimization. The selected geometries, RAE 2822 airfoil and DPW-w1 wing, were parameterized with volumetric NURBS.

This work is within the aerodynamic shape design and optimization research line of INTA's Fluid Dynamics Branch.

Surrogate-Based Optimization Strategy

This section introduces each of the components of the SBO approach applied in this study: geometry parameterization through volumetric NURBS, Evolutionary Algorithms (EAs), Support Vector Machines for Regression (SVR) and the Intelligent Estimation Search with Sequential Learning (IES-SL) as the strategy for adaptive sampling focused on optimization.

Geometry Parameterization

Parameterization is a crucial step in an aerodynamic design optimization problem. NURBS have demonstrated to be able to accurately represent a large family of geometries. In aerodynamic design, NURBS provide smooth surfaces while maintaining some deformation locality (Mousavi et al. 2007). In addition, the optimized surface at the end of the optimization process has the correct format to feed directly the CAD and grid generation applications. However, the use of surface NURBS can be impractical, because very frequently requires the additional effort to develop a surface representation that fits the original geometry, with an appropriated arrange of control points for the optimization. An alternative approach is to envelop the geometry in a volumetric NURBS (Martin et al. 2013), which maintain the deformation properties of a conventional 2-dimensional surface, but with the advantage that control points can be set up arbitrarily.

From a mathematical point of view, NURBS surfaces are defined as the tensor product of three NURBS curves, defining a volumetric region, where the deformation is governed by the movement of control points:

$$S(\xi, \eta, \mu) = \frac{\sum_i^I \sum_j^J \sum_k^K U_{i,n}(\xi) V_{i,n}(\eta) W(\mu) C_{ijk}}{\sum_i^I \sum_j^J \sum_k^K U_{i,n}(\xi) V_{i,n}(\eta) W(\mu)} \quad (1)$$

where C are the control points, ξ , η , and μ are the parametric coordinates, and U , V , and W are the basis functions which are calculated using the following expression:

$$U_{i,1}(\xi) = \begin{cases} 1 & \text{if } u_i \leq \xi < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

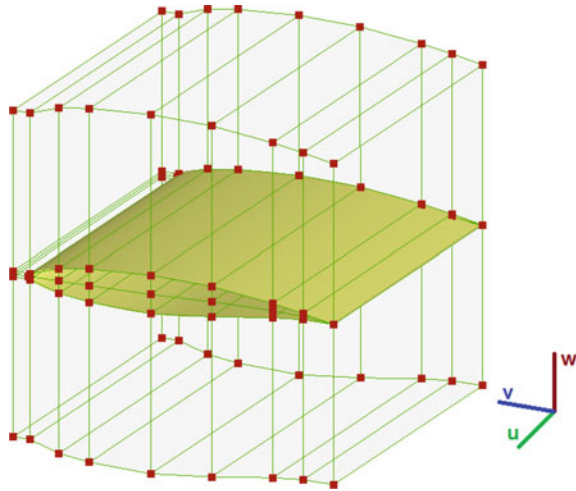
$$U_{i,k}(\xi) = \frac{(\xi - u_i)U_{i,k-1}(\xi)}{u_{i+k-1} - u_i} + \frac{(u_{i+k} - \xi)U_{i+1,k-1}(\xi)}{u_{i+k} - u_{i+1}} \quad (2)$$

The basis coefficients are calculated from the knot vectors \bar{U} , \bar{V} and \bar{W} , and, which are a sequence of real numbers. Basis functions are equal to zero everywhere except for an interval delimited by the order of the NURBS, defining the area of influence of each control point (Piegl and Tiller 1997). The most common implementation of the control box is to employ uniform basis, which can be obtained with a knot sequence as:

$$\left\{ \underbrace{0, \dots, 0}_{p+1}, \frac{1}{N}, \dots, \frac{i}{N}, \dots, \frac{N-1}{N}, \underbrace{1, \dots, 1}_{p+1} \right\} \quad (3)$$

First order is equivalent to a linear interpolation, while second and third orders provide derivative and curvature continuity, respectively.

Fig. 1 RAE 2822 control box parameterization



In this work, the airfoil is parameterized with third order volumetric NURBS, also called control box, and the design variables will be the vertical displacements (z axis) of the 14 control points. Figure 1 depicts the selected parameterization.

To clarify, there are additional control points at the trailing and leading edge that are kept fixed, in order to maintain the angle of attack; so these control points are not considered as design variables.

Evolutionary Algorithm

Evolutionary algorithms (EAs) are bio-inspired methods that clone the behaviour of natural evolution to solve complex optimization problems. The basic elements of an EA are the solution coding, the selection operator and the crossover and mutation operator.

In the design application to be considered in this work, each coding vector is composed by a given parameterization of a geometry, i.e., $z = [cp1, cp2, cp3, \dots, cpN]$, where cp is the vertical coordinates of each control point.

More details about the EA applied in this paper can be found in a previous work from the authors (Andrés et al. 2012).

Objective Function Approximation Using Support Vector Machines (SVMs)

Support vector machines acts as a meta-model to predict the objective function to be optimized, which in this case is given by the aerodynamic performance of de airfoil.

Support Vector machines for Regression are a powerful tool used on the machine learning field, and a modelling tool for a large amount of regression problems on engineering. The SVR can be solved as a convex optimization problem using kernel theory to face nonlinear problems. The SVR consider not only the prediction error but also the generalization of the model. To obtain the best performance, a search of the most suitable combination of the kernel parameters must be carried on, usually by using cross validation techniques over the training set. To reduce the computational time of this process, different methods have been proposed in the literature to reduce the search space related to these parameters. In this case, it has been applied the one developed by Ortiz-García et al. (2009). Which has proven to require pretty short search times.

More details about the EA applied in this paper can be found in a previous work from the authors (Andrés et al. 2012)

Flowchart of the Proposed Approach

In this article, The Intelligent Estimation Search with Sequential Learning (IES-SL) method is applied. This method allows performing an efficient adaptive sampling guiding the optimization algorithm towards the most promising regions of the design space. The flowchart of the proposed approach is depicted in Fig. 2. First, an initial set of randomly generated (including the baseline) geometries are selected and evaluated with CFD tool (DLR Tau code in this work). With this set, a first surrogate is built and linked within an evolutionary algorithm. The latter will search for the minimum of the surrogate in each of the optimization iterations, and the returned optima will be again evaluated using the high-fidelity CFD solver, and then incorporated to the surrogate model, which is rebuilt and more precise on each iteration. The process will end when a certain number of CFDs budget is reached.

The aim of this work is to study the influence of the initial training size in the precision of the surrogate and the convergence of the proposed approach.

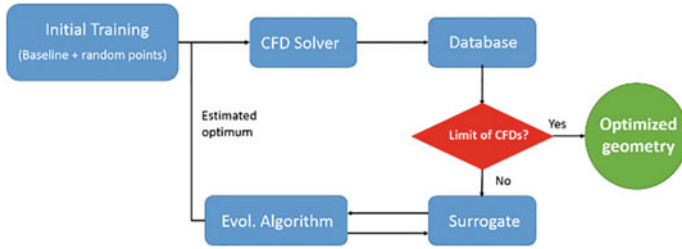


Fig. 2 Flowchart of the proposed approach

Table 1 Baseline airfoil features

Chord (m)	0.61
Maximum thickness-to-chord ratio	0.0121 at $x/c = 0.38$
Maximum camber-to-chord ratio	0.0126 at $x/c = 0.76$
Leading edge radius (m)	0.00827
Airfoil area (m ²)	0.0776
Trailing edge angle	9°

Numerical Results

Baseline Geometry

The selected geometry for this study was the well-known RAE2822 airfoil features described in Table 1. The airfoil is a rear-loaded, sub-critical geometry, designed to exhibit a roof-top type pressure distribution at design conditions ($Mach = 0.66$, $Cl = 0.56$ ESDU 1973). It has been tested in the RAE wind tunnel in 11 different flow conditions in the range of Mach numbers from 0.676 to 0.750 and at several Reynolds numbers (Cook et al. 1979).

A 56 k points unstructured grid was generated for this study.

Test Case Definition

The proposed approach is applied to 5 optimizations cases with 4, 8, 16, 32 and 64 initial random training points respectively. The multipoint optimization problem of the RAE 2822 is selected. The flow conditions for both design points 1 & 2 are:

	DP1	DP2
Mach	0.734	0.754
Re	6.5 M	6.2 M
Turb. Model	SA	$\kappa\omega$ TNT

Table 2 OF evolution respect the initial training size

# Initial training random points	Objective function (OF)
4	0.6014
8	0.6059
16	0.6021
32	0.6016
64	0.5993

The objective function selected was $Min\left(\frac{C_D}{C_L}\right)$ with some considerations. These are:

- Aerodynamics constraints and penalties:
 1. Prescribed minimum lift coefficient: $C_l^0|_k : C_l|_k \geq C_l^0|_k$.
 2. Prescribed minimum pitching coefficient: $C_m^0|_k : C_m|_k \geq C_m^0|_k$.
 3. Drag penalty: if constraint on minimum pitching moment is not satisfied, the penalty will be 1 drag count per 0.01 in ΔC_m .
- Geometric constraints
 1. Limit: $\pm 20\%$ of the initial control points' values.
 2. Prescribed maximum thickness ratio $(t/c)_{\max} : \max(t/c) = (t/c)_{\max}$.
 3. Prescribed minimum thickness ratio $(t/c)_{\min}^{80}$ at $x = 0.8c$: $(t/c)^{80} \geq (t/c)_{\min}^{80}$.
 4. Prescribed minimum leading edge nose radius $R_{\min}^{le} : R^{le} \geq R_{\min}^{le}$.

Sensitivity Study Results

In this section, the results of the present study are presented. Three issues are analysed. First, the influence of the initial training size in the convergence of the method. Next, the influence in the method precision of the initial data set. Finally, the value of the objective function reached in each case.

Regarding the first analysis, Fig. 3 shows the convergence of the IES-SL for each test case. As can be seen, the five test cases have a huge oscillation during the “training period”. This is the expected behaviour since the points in this data set are generated randomly. A lower size of initial training means the optimizer requires more iterations to reach the “optimum region”. The reason is that the initial surrogate is more intelligent with a huge initial data set, but it requires more time to be built. At last, the five test cases reach the same optimum region (see Table 2).

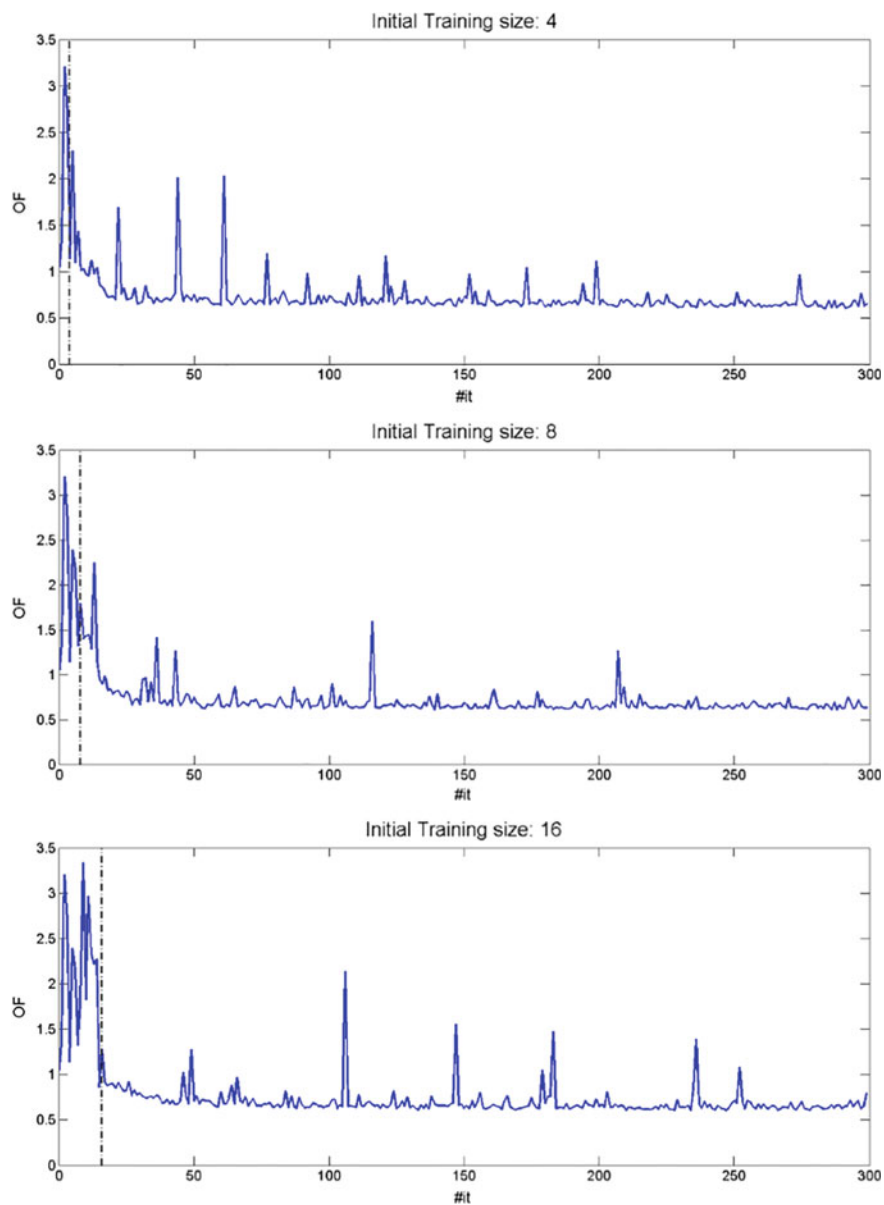


Fig. 3 SBGO convergence versus initial data set size

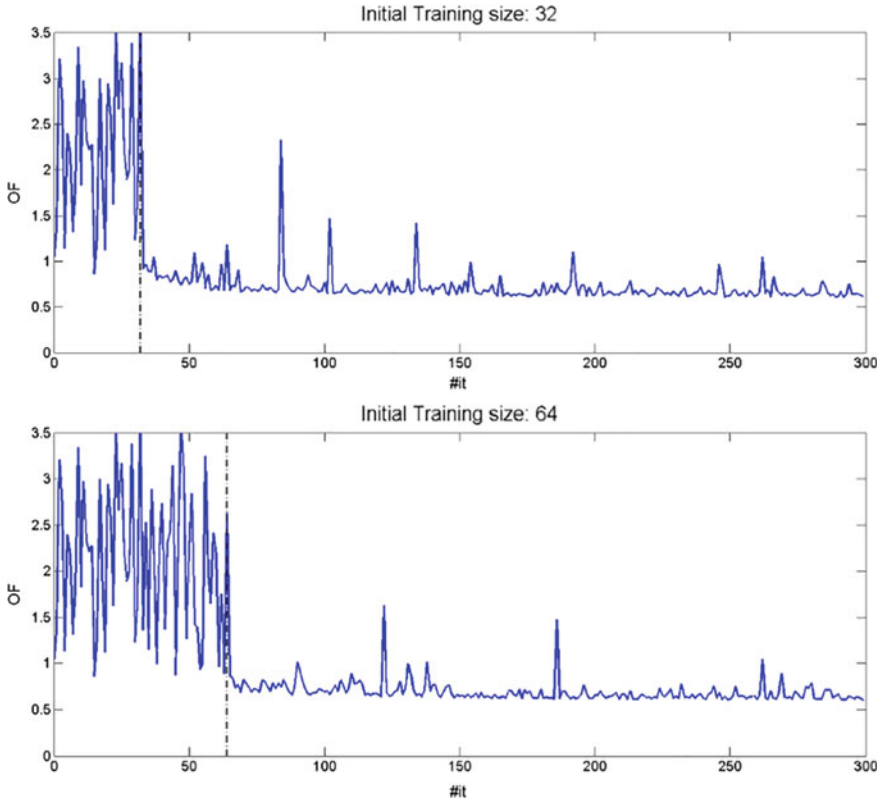


Fig. 3 (continued)

Figure 4 illustrates the accuracy of the method with respect the initial training size. As expected, an initial surrogate with a vast number of points has an initial accuracy higher than one with a small set of points. This is in the same line that the convergence. Nevertheless, it requires more time to start the optimum seek.

Last, but not least, Table 2 summarizes the value of the OF reached in each case. It can be seen that there is no influence of the initial training size in the final value of the OF (with a reasonable budget of iterations). This is the main advantage of the IES-SL proposed.

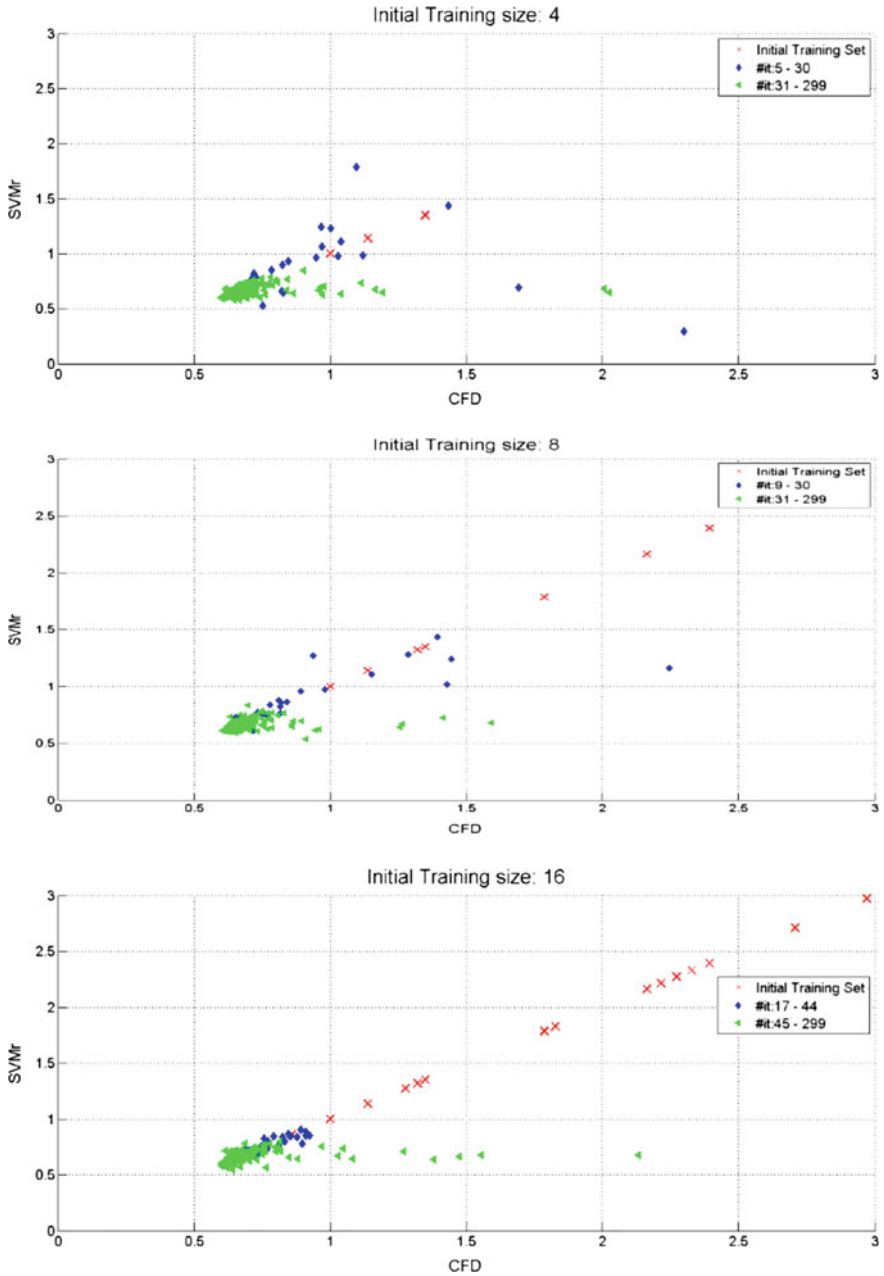


Fig. 4 Approach accuracy for each initial training size

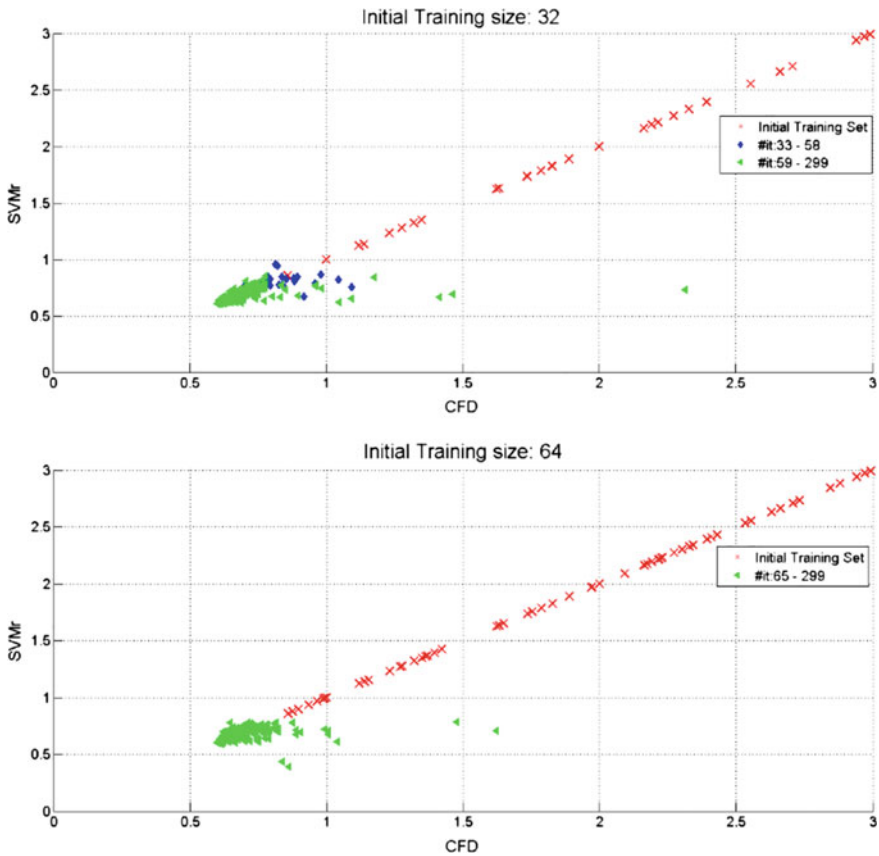


Fig. 4 (continued)

Conclusions

The aim of this work was to provide an analysis about how the initial training size of the surrogate affects the behaviour of the proposed IES-LS method. The following conclusions have been extracted from the solutions:

- The optimum region reached is the same independently the training set size. A model with higher initial data set size requires less iterations to reach de optimum region, but it requires more computational time to be built, which is not feasible from the industry point of view.
- In the same trend, the initial accuracy of the surrogate increases with the number of training samples, but the drawback is the same which is exposed in the previous point.
- As summarized in Table 2, the training set size has no influence in the OF reached by the proposed IES-SL approach.

In summary, the main advantage of the proposed SBO method is that it can reach the global optimum with a small number of initial samples. This is feasible due to sequential learning allows the surrogate to become accurate each iteration. So, there's an important reduction of the initial computational cost that requires a standard offline SBO.

Acknowledgements The research described in this work has been supported under INTA activity "Termofluidodinámica" (IGB99001) and the 'Rafael Calvo Rodés' scholarship.

References

- Andrés E, Salcedo-Sanz S, Mongue F, Pérez-Bellido A (2012) Efficient aerodynamic design through evolutionary programming and support vector regression algorithms. *Expert Syst Appl* 39:10700–10708
- Andrés-Pérez E, Iuliano E (2015) Application of surrogate-based global optimization to aerodynamic design. *Springer Tracts in Mechanical Engineering*
- Andrés-Pérez E, González-Juárez D, Martín-Burgos M, Carro-Calvo L, Salcedo-Sanz S (2016) Influence of the number and location of design parameters in the aerodynamic shape optimization of a transonic airfoil and a wing through evolutionary algorithms and support vector machines. *Eng Optim* 48
- Cook P, McDoland M, Firmin M (1979) Airfoil RAE 2822 - Pressure Distributions, and Boundary Layer and Wake Measurements. AGARD Report 138
- ESDU (1973) Second-order method for estimating the subcritical pressure distribution on a two-dimensional airfoil in compressible inviscid flow
- González-Juárez D, Andrés-Pérez E, Martín-Burgos M, Carro-Calvo L, Salcedo-Sanz S (2015) Influence of geometry parameterization in aerodynamic shape design of aeronautical configurations by evolutionary algorithms. In: 6th European conference for aeronautics and space sciences (EUCASS). Krakow, Poland
- Iuliano E, Quagliarella D (2013) Aerodynamic shape optimization via non-intrusive POD-based surrogate modeling. In: IEEE congress on evolutionary computation. Cancún, Mexico
- Jahangirian A, Shahrokhi A (2011) Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured CFD solver. *Comput Fluids* 46:270–276
- Keane A (2003) Wing optimization using design of experiment, response surface, and data fusion methods. *J Aircr* 40(4):741–750
- Koziel S, Leifsson L (2013) Multi-level surrogate-based airfoil shape optimization. In: 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, grapevine (Dallas/Ft. Worth Region), Texas
- Leifsson L, Koziel S, Tesfahunen Y (2014) Aerodynamic design optimization: physics-based surrogate approaches for airfoil and wing design. In: AIAA SciTech
- Li C, Brezillon J, Görtz S (2001) A framework for surrogate-based aerodynamic optimization. In: ONERA-DLR aero-space symposium, ODAS
- Likeng H, Zhenghong G (2012) Wing-body optimization based on multi-fidelity surrogate model. In: International congress of the aeronautical sciences ICAS. Brisbane, Australia
- Lukaczyk T, Palacios F, Alonso J (2014) Active subspaces for shape optimization. In: AIAA SciTech
- Martin M, Andrés E, Valero E, Lozano C (2013) Gradients calculation for arbitrary parameterizations via volumetric NURBS: the control box approach. In: EUCASS
- Mousavi A, Castonguay P, Nadarajah S (2007) Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization. In: AIAA computational fluid dynamics. Miami

- Ortiz-García E, Sanz SS, Pérez-Bellido ÁM, Portilla-Figueras JA (2009) Improving the training time of support vector regression algorithms through novel hyper-parameters search space reductions. *Neurocomputing*
- Parr J, Holden C, Forrester A, Keane A (2010) Review of efficient surrogate infill sampling criteria with constraint handling. In: 2nd international conference on engineering optimization. Lisbon, Portugal
- Piegl L, Tiller W (1997) *The NURBS book*. Springer, Berlin

Garteur AD/AG-52: Surrogate-Based Global Optimization Methods in Preliminary Aerodynamic Design



Esther Andrés-Pérez, Daniel González-Juarez, Mario Martin, Emiliano Iuliano, Davide Cinquegrana, Gerald Carrier, Jacques Peter, Didier Bailly, Olivier Amoignon, Petr Dvorak, David Funes, Per Weinerfelt, Leopoldo Carro, Sancho Salcedo, Yaochu Jin, John Doherty and Handing Wang

Abstract This work presents a summary of the results obtained during the activities developed within the GARTEUR AD/AG-52 group. GARTEUR stands for “Group for Aeronautical Research and Technology in Europe” and is a multinational organization that performs high quality, collaborative, precompetitive research in the field of aeronautics to improve technological competence of the European Aerospace Industry. The aim of the AG52 group was to make an evaluation and assessment of

E. Andrés-Pérez (✉)

Engineering Department, Ingeniera de Sistemas para la Defensa de España S.A. (ISDEFE-INTA), Madrid, Spain
e-mail: eandres@isdefe.es

E. Andrés-Pérez

Engineering Department, Universidad Politécnica de Madrid (UPM), Madrid, Spain

D. González-Juarez · M. Martin

Fluid Dynamics Branch, National Institute for Aerospace Technology (INTA), Madrid, Spain
e-mail: gonzalezjd@inta.es

M. Martin

e-mail: martinbj@inta.es

E. Iuliano · D. Cinquegrana

Fluid Dynamics Lab, Centro Italiano Ricerche Aerospaziali (CIRA), Capua, Italy
e-mail: e.iuliano@cira.it

D. Cinquegrana

e-mail: d.cinquegrana@cira.it

G. Carrier · J. Peter · D. Bailly

The French Aerospace Lab (ONERA), Palaiseau, France
e-mail: Gerald.Carrier@onera.fr

J. Peter

e-mail: jacques.peter@onera.fr

D. Bailly

e-mail: Didier.Bailly@onera.fr

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_13

surrogate-based global optimization methods for aerodynamic shape design of aeronautical configurations. The structure of the paper is as follows: Sect. “[Introduction](#)” will introduce the state-of-the-art in surrogate-based optimization for aerodynamic design and Sect. “[Definition of Common Test Cases and Methods](#)” will detail the test cases selected in the AG52 group. Optimization results will be then showed in Sect. “[Optimization Results](#)”, and conclusions will be provided in the last section.

Introduction

The AD/AG 52 has been established to explore and unveil the potential of surrogate-based techniques in aerodynamic shape optimization. Any designer has experienced the burden of intensive numerical optimization involving CFD or analogous expensive black-box simulations. Typically, the computational load is well tolerated when dealing with two-dimensional airfoil shape design. However, the order of magnitude of both the number of simulations required and the CPU time for single evaluation grows significantly with the increase of the dimensionality (e.g., from two-dimensional to three-dimensional cases) and of the inherent geometric complexity (e.g., wing-fuselage configuration, high-lift cases, wing-pylon-engine junction) of

O. Amoignon

Swedish Defence Research Agency (FOI), Kista, Sweden
e-mail: olivier.g.amoignon@gmail.com

P. Dvorak

Aerodynamics and Space Technology Group,
Brno University of Technology VUT, Brno, Czech Republic
e-mail: dvorak.p@fme.vutbr.cz

D. Funes

AIRBUS- Military, Madrid, Spain
e-mail: david.e.funes@airbus.com

P. Weinerfelt

SAAB Aeronautics, Linköping, Sweden
e-mail: Per.Weinerfelt@saabgroup.com

L. Carro · S. Salcedo

Department of Signal Theory and Communications,
Universidad de Alcalá UAH, Alcalá de Henares, Spain
e-mail: leopoldo.carro@uah.es

S. Salcedo

e-mail: sancho.salcedo@uah.es

Y. Jin · J. Doherty · H. Wang

University of Surrey UNIS, Guildford, UK
e-mail: yaochu.jin@surrey.ac.uk

J. Doherty

e-mail: john.doherty@surrey.ac.uk

the problem at hand. Surrogate models are able to complement, not to replace, the “true” function evaluation by providing a fast and adaptive response during screening parametric analyses and numerical optimization. Building and querying a surrogate is a way to potentially acquire new information about the problem under analysis, not to directly solve it: indeed, any surrogate, even the smartest ones, have to face with the prediction error and minimize it in order to be accurate. This does not hamper the usefulness of the approach as, even in presence of errors away from the sampled data, function trends and optimization directions can be caught to enrich the process.

The main objective of this Action Group was to make a deep evaluation and assessment of surrogate-based global optimization methods for aerodynamic shape optimization.

The work structure for this AG is application-driven, and it was composed of 2 tasks. First, in task 1, two common test cases were proposed and were addressed by all partners using different methods. The objective was to make an exhaustive comparison of promising methods and quantification of their performance in terms of accuracy and CPU cost. Then, in task 2, more industry-relevant test cases were provided, and the consortium have used the knowledge acquired in task 1, to solve such test cases.

State-of-Art

Global search methods are traditionally based on stochastic optimization techniques; most of them are population-based whereas there are few individual-based algorithms. The most commonly used population-based methods are the Evolutionary Algorithms (EAs; including Genetic Algorithms-GAs and Evolution Strategies-ES). However, other alternatives, such as Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995) or Bacterial Foraging Optimization (BFO) (Muller et al. 2002), Differential Evolution (DE) (Storn and Price 1997) etc., exist. Evolutionary algorithms (EAs) (Duvigneau and Visonneau 2004) are successful single- and multi-objective constrained optimization methods that can handle any kind of objective function and may accommodate any evaluation software as a black-box tool. Due to the high and expensive number of required calls, EAs assisted by surrogate evaluation models (metamodels) have been devised and, depending on the training method, they can be classified into off-line trained metamodels (Jin et al. 2002; Buche et al. 2005; Won and Ray 2005), or on-line trained metamodels (Giannakoglou 2002; Branke and Schmidt 2005; Jin 2005; Szollos et al. 2009).

There are different kinds of surrogate modelling as for example Polynomial Regression (PR), Multivariate Adaptive Regression Splines (MARS), Gaussian Processes, Kriging (KG), Cokriging (Zhong-Hua et al. 2010), Artificial Neural Networks (ANN) (de Weerd et al. 2005; Marinus et al. 2010), Radial Basis Functions (RBF)

(Praveen and Duvigneau 2007), Proper Orthogonal Decomposition (POD) methods (Iuliano and Quagliarella 2013) and Support Vector Machines (SVM) (Clarke et al. 2005; Andres et al. 2011), among others. A reference for recent advances in surrogate based optimization techniques can be found in (Forrester and Keane 2009), and comparison of surrogate models for turbomachinery design in (Peter and Marcelet 2007). Also, surrogate modelling has been already applied for the design optimization of composite aircraft fuselage panels (Vankan and Maas 2010). In addition, the use of Kriging surrogate model in combination with evolutionary algorithms has been recently applied for the design of hypersonic vehicles (Ahmed and Qin 2010). Furthermore, the use of Support Vector Regression algorithms (SVMr) as metamodels has been applied to a large variety of regression problems, in many of them mixed with evolutionary computation algorithms (Cheng et al. 2011; Salcedo-Sanz et al. 2011; Jiang and He 2012; Iuliano and Andrés 2015).

Current research focuses on the improvement of metamodels (by using artificial neural networks, Gaussian models, etc., or proposing metamodels variants (Younis et al. 2008) based on not only the responses but also the gradient or responses, Kriging) and/or different metamodels implementation schemes within the Metamodel-Assisted Evolutionary algorithm (MAEA) (Lim et al. 2010). Particular attention is required in multi-objective optimization problems, where a Pareto front of non-dominated solutions is sought and the evolving individuals are dispersed in the design space, or when asynchronous MAEAs (Asouti and Giannakoglou 2009; Kapsoulis et al. 2016) are devised by overcoming the notion of generation and the corresponding synchronization barrier.

With respect to the combination of global and local search methods within the design optimization process, the so-called hierarchical approach has been proposed in the literature (for instance, stochastic methods for the exhaustive search of the design space along with gradient-based methods for the refinement of promising solutions) (Kampolis and Giannakoglou 2011; Peter et al. 2011). Metamodel-assisted memetic algorithms (Kapsoulis et al. 2016) are also hybrid schemes that combine the use of global and local optimization methods (Carrier 2006; Bompard et al. 2010; Leifsson et al. 2014).

Definition of Common Test Cases and Methods

Two test cases had been selected to assess and compare methods: the RAE 2822 airfoil and the Drag Prediction Workshop (DPW) W1 wing. The first is two-dimensional and it has been widely studied in the aerospace community over the last decades; a fair number of both experimental and numerical data exist as well as optimization results collected with a variety of methodologies. Transonic viscous flow conditions were considered for this test case.

The second test case was taken as the DPW-W1 wing which has been proposed during the 3rd AIAA Drag Prediction Workshop (<http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3/workshop3.html>): it is a quite simple wing geometry that can



Fig. 1 RAE2822 baseline geometry

Table 1 Baseline airfoil features

Chord (m)	0.61
Maximum thickness-to-chord ratio	0.121@ $x/c = 0.38$
Maximum camber-to-chord ratio	0.0126@ $x/c = 0.76$
Leading edge radius (m)	0.00827
Airfoil area (m ²)	0.0776
Trailing edge angle	9°

be easily handled in an optimization context. Again, experimental and numerical data are available for comparison. Transonic viscous flow conditions were also considered.

RAE2822 Airfoil

The RAE 2822 airfoil (Cook et al. 1979) had been selected as the initial geometry for aerodynamic optimizations. The airfoil contour shape is shown in Fig. 1 and Table 1 summarizes its geometrical characteristics.

The flow conditions and constraints of different design points were the inputs for the optimization process. These flow conditions included prescribed angle of attack (AoA), Mach number, Reynolds number as it is shown:

- DP1 (Case 9): $M = 0.734, Re = 6.5 \times 10^6, AoA = 2.65^\circ$
- DP2 (Case 10): $M = 0.754, Re = 6.2 \times 10^6, AoA = 2.65^\circ$

The objective function defined was to maximize lift over drag ratio at both the design points, while maintaining some specified constraints.

The aerodynamic constraints and penalties considered were:

- i. Prescribed minimum lift coefficient $\rightarrow C_l^0|_k : C_l|_k \geq C_l^0|_k$.
- ii. Prescribed minimum pitching moment coefficient $\rightarrow C_m^0|_k : C_m|_k \geq C_m^0|_k$
 where $C_l^0|_k$ and $C_m^0|_k$ are the lift and pitching moment coefficients, respectively, of the initial geometry, for the design point k.
- iii. Drag penalty: if constraint on minimum pitching moment is not satisfied, the penalty will be 1 drag count per 0.01 in ΔC_m .

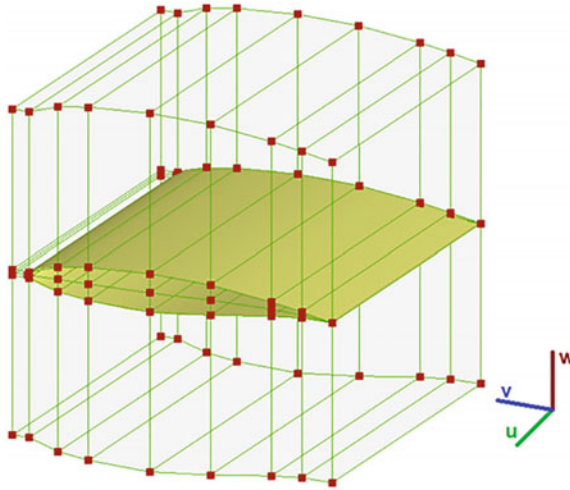


Fig. 2 NURBS control box

While the geometric constraints were:

- i. Prescribed maximum thickness ratio $(t/c)_{\max}$: $\max(t/c) = (t/c)_{\max}$
- ii. Prescribed minimum thickness ratio $(t/c)_{\min}^{80}$ at $x = 0.8c$: $(t/c)^{80} \geq (t/c)_{\min}^{80}$
- iii. Prescribed minimum leading edge nose radius R_{\min}^{le} : $R^{le} \geq R_{\min}^{le}$

The RAE2822 was parameterized by a volumetric NURBS. Figure 2 shows the parameterization in green color, with the control points marked in red. The selected parameterization is a 3D control box with 2 control points in direction u (fake 3D grid), 10 in direction v and 5 in direction w .

DPW Wing

The public domain DPW-W1 wing (<http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3/workshop3.html>; Epstein et al. 2008) was selected as the initial geometry for aerodynamic optimizations. Reference quantities for this wing are displayed in Table 2 while Fig. 3 depicts the geometry

The flow conditions and constraints of different design points were the inputs for the optimization process. These flow conditions included prescribed cruise lift, Mach number, Reynolds number as it is shown:

- $M = 0.76, C_L = 0.5, Re = 5 \times 10^6$ DP1 (main design point)
- $M = 0.78, C_L = 0.5, Re = 5 \times 10^6$ DP2 (high-Mach design point).

The design goal was to achieve a geometry with the minimum drag, while maintaining some specified aerodynamics and geometric constraints.

Table 2 Reference quantities for the DPW wing

S_{ref} (wing reference area)	290322 mm ²
C_{ref} (wing reference chord)	197.55 mm
X_{ref}	154.24 mm (relative to the wing root leading edge)
b/2 (semi span)	762 mm
AR (aspect ratio, $AR = b^2/S_{ref}$)	8.0

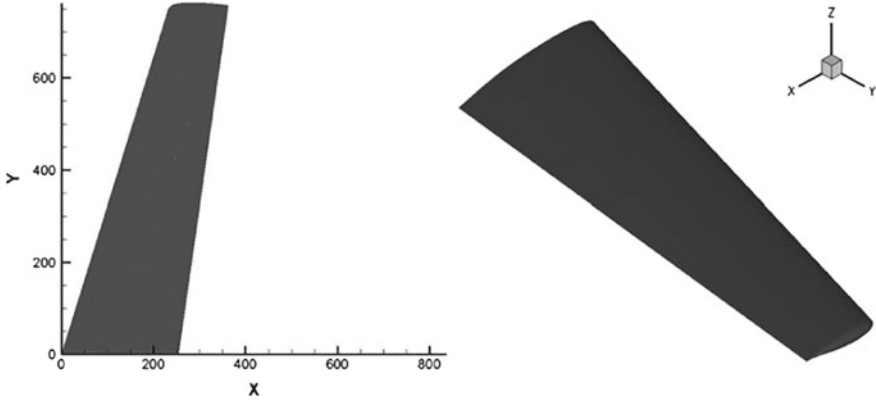


Fig. 3 Planform plot of the initial geometry (left), 3D plot of the initial geometry (right)

In this case, the aerodynamic constraints and penalties taking account were:

- i. Prescribed constant lift coefficient: $C_L^0 \rightarrow C_L(k) = C_L^0(k)$.
- ii. Minimum pitching moment: $C_M^0 \rightarrow C_M(k) \geq C_M^0(k)$
 $C_L^0(k)$ and $C_M^0(k)$ are the lift and pitching moment coefficients, respectively, of the initial geometry, for the design point k .
- iii. Drag penalty: If constraint in minimum pitching moment is not satisfied, the penalty will be 1 drag count per 0.01 in ΔC_M .

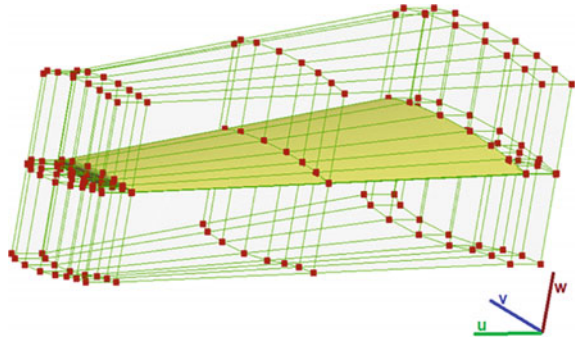
while the geometric constraints were:

- i. Airfoils' maximum thickness constraints: $(t/c)_{section} \geq (t/c)_{section}^0$
 where $(t/c)_{section}^0$ is the maximum thickness for the original wing sections, root, mid-span and tip: $(t/c)_{root}^0 = (t/c)_{mid-span}^0 = (t/c)_{tip}^0 = 13.5\%$. Therefore, the maximum thickness for the optimized wing sections' should be greater or equal than 13.5%.
- ii. Beam constraints:
 First, two locations (x/c) are fixed to represent the beam constraints:

$$(x/c)_{root,1} = (x/c)_{mid-span,1} = (x/c)_{tip,1} = 0.20$$

$$(x/c)_{root,2} = (x/c)_{mid-span,2} = (x/c)_{tip,2} = 0.75$$

Fig. 4 DPW geometric constraints in parameterization



Then, the thickness value of the original wing sections at these locations are defined by:

$$\begin{aligned} (t/c)_{root,1}^0 &= (t/c)_{mid-span,1}^0 = (t/c)_{tip,1}^0 = 12\% \\ (t/c)_{root,2}^0 &= (t/c)_{mid-span,2}^0 = (t/c)_{tip,2}^0 = 5.9\% \end{aligned}$$

The parameterization defined for task 2 is depicted in Fig. 4. The DPW wing was parameterized by a 3D control box with 5 control points in direction u , 10 in direction v and 5 in direction w . The parametric u direction corresponds to the y axis, the v direction to the x axis, and the w direction to the z axis. The design variables to be modified are the control points in the w direction.

Applied Approaches

The surrogate models employed by the partners are listed in Table 3.

Table 3 Summary of test cases in Task 1 and 2 and involved partners and methods

	TC1.1 RAE2822 airfoil (RANS)	TC1.2 DPW-W1 wing (Euler)	TC1.2 DPW-W1 wing (RANS)
INTA/UAH	SVMs	SVMs	SVMs
VUT	ANNs	–	–
CIRA	POD/RBF, Kriging/EGO	–	Kriging/EGO
FOI	Kriging, RBF	–	–
ONERA	Kriging	–	–
UNIS	Ensemble	–	–
Airbus-M	–	–	HOSVD

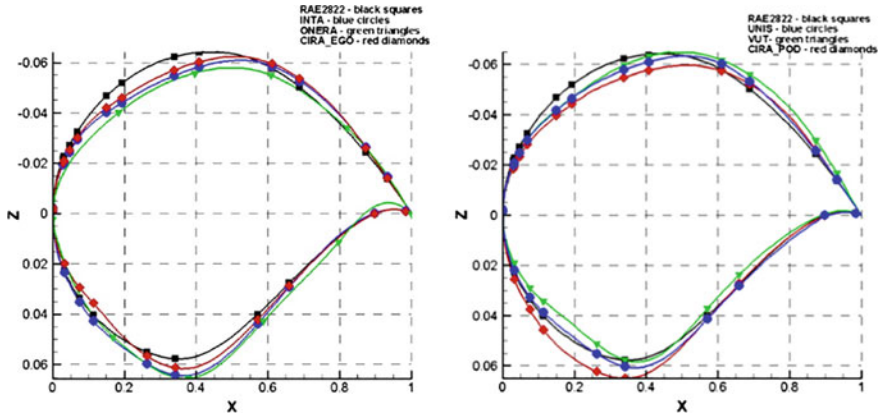


Fig. 5 Comparison of partners’ optimized geometries and baseline RAE2822

Optimization Results

Task 1 RAE2822 RANS

Partner’s optimized shapes are depicted in Fig. 5 while Fig. 6 shows the Cp distributions for DP1 and DP2. The objective function values obtained for each geometry after the cross validation with several solvers are summarized in Table 4.

Table 4 Average value of the objective function values for RAE 2822 in viscous flow conditions optimization

	Mean OF (TAU, MSES, ZEN 3 levels)	Mean OF (only TAU and ZEN fine)
RAE 2822 baseline	1	1
CIRA-POD	0.6223	0.6266
CIRA-EGO	0.6208	0.6236
INTA/UAH	0.6243	0.6211
ONERA	0.6494	0.6498
UNIS	0.6367	0.6338
VUT	0.6969	0.7063

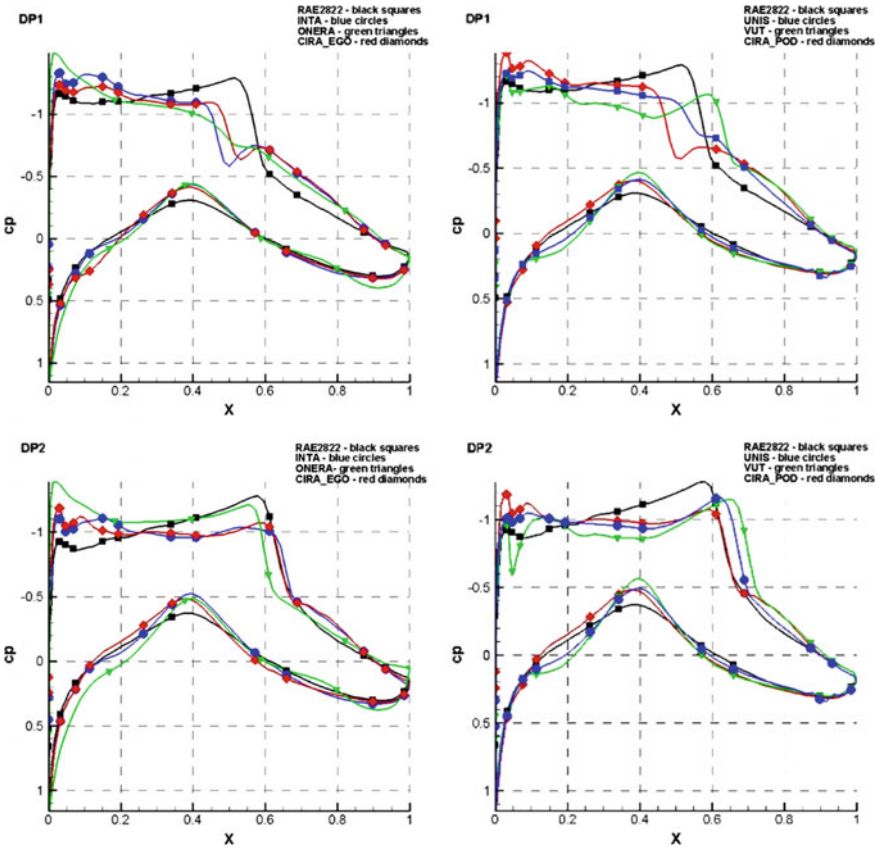


Fig. 6 Cp distribution for both DP1 and DP2 for different geometries

Task 2 DPW-W1 RANS

Partner’s optimized shapes at 25, 50 and 75% wingspan are depicted in Fig. 7 and corresponding Cp distributions are showed in Figs. 8 and 9. The objective function values obtained for each geometry after the cross validation with several solvers are summarized in Table 5.

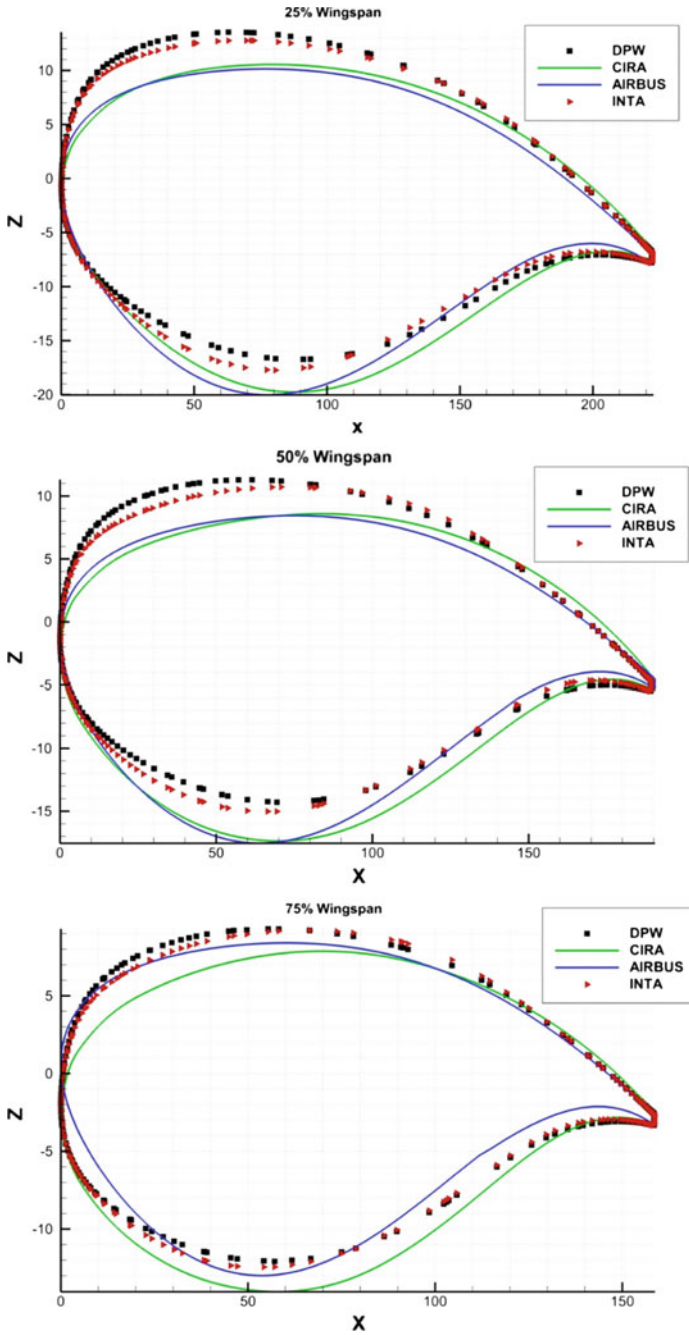


Fig. 7 Comparison of partners' optimized geometries and baseline DPW-w1. 25, 50 and 75% wingspan

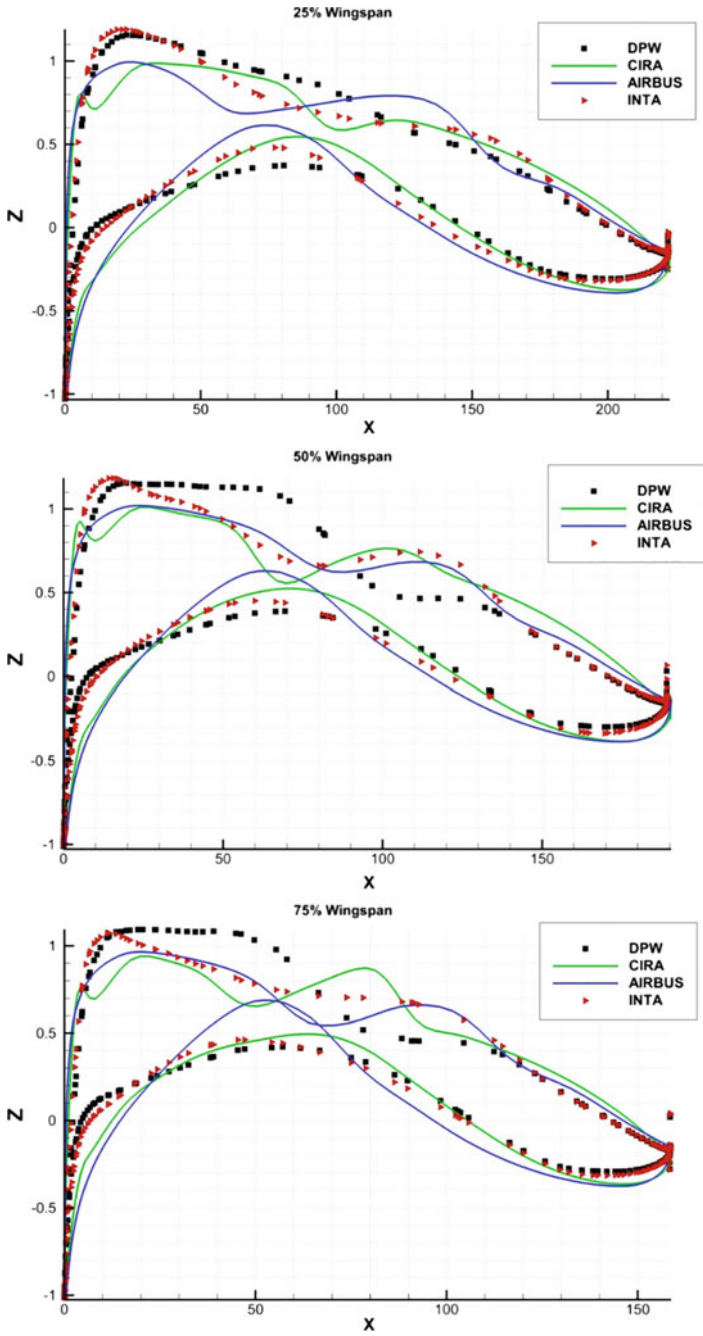


Fig. 8 Comparison of C_p distributions for DP1 at 25, 50 and 75% wingspan

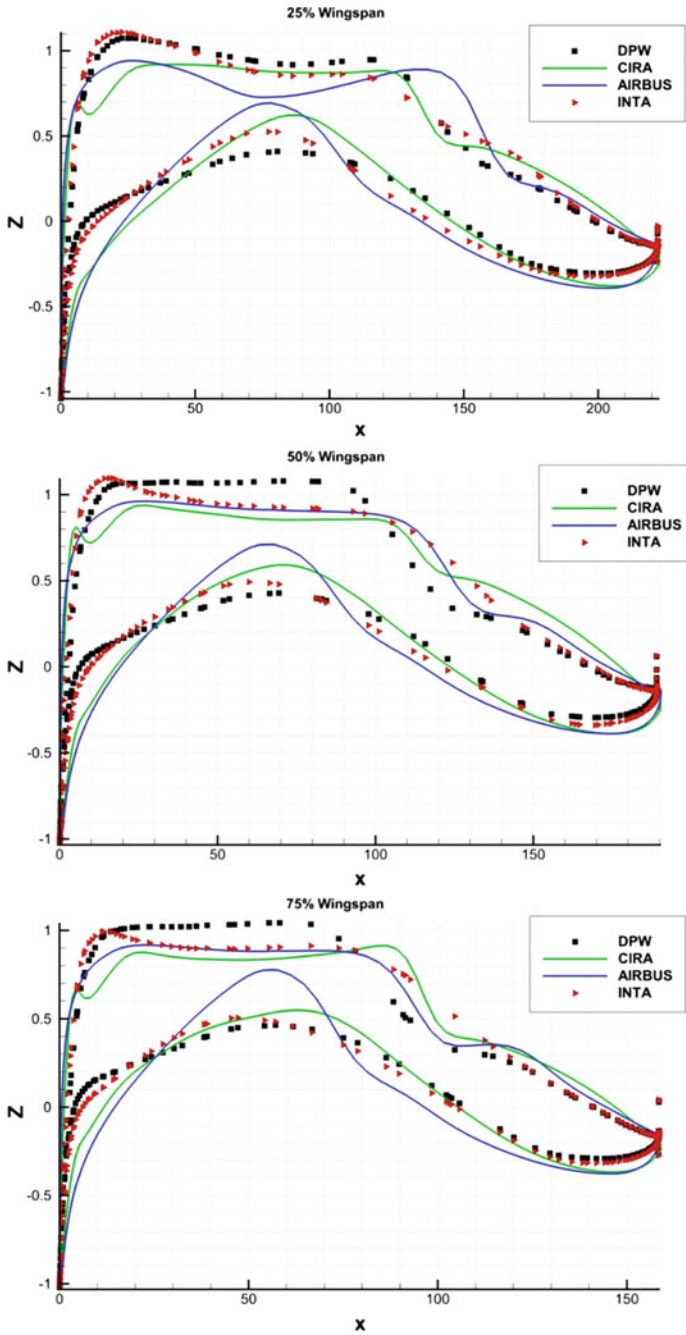


Fig. 9 Comparison of Cp distributions for DP2 at 25, 50 and 75% wingspan

Table 5 Results of cross-analysis of the optimized geometries ZEN and TAU solvers

Geometry	Solver	CL	CD	CM	CL	CD	CM	OF
DPW w1	ZEN	0.5	0.0237	-0.069	0.5	0.0264	-0.078	1
DPW w1	TAU	0.5	0.0237	-0.067	0.5	0.0267	-0.070	1
CIRA	ZEN	0.5	0.0224	-0.084	0.5	0.0232	-0.089	0.91
CIRA	TAU	0.5	0.0221	-0.075	0.5	0.0233	-0.079	0.91
INTA/UAH	ZEN	0.5	0.0235	-0.084	0.5	0.0241	-0.091	0.96
INTA/UAH	TAU	0.5	0.0231	-0.074	0.5	0.0248	-0.077	0.94
AIRBUS-M	TAU	0.5	0.0231	-0.090	0.5	0.0238	-0.078	0.92

Conclusions

This paper summarized the results of the GARTEUR AD/AG52 group on “Surrogate-based global optimization methods for aerodynamic design”.

Surrogate-based global optimization has been demonstrated to be feasible for aerodynamic design in case of high number of design variables (tested 36 on DVs).

However, the accuracy of the surrogate models strongly depends on the sampling and the objective of the surrogate:

- If the objective is to provide general predictions, an a priori LHS sampling in combination or not with Lola-Voronoi sampling seems to be a good option.
- If the objective is to better predict those regions of the design space where the optimum is located, then a mixed a priori and adaptive sampling is recommended.

In case of optimization best results were achieved by the adaptive Kriging, HOSVD-based and SVMr optimization approaches.

Interested readers can consult the complete AG52 report in www.garteur.org.

References

- Ahmed M, Qin N (2010) Metamodels for aerothermodynamic design optimization. In: 48th AIAA aerospace sciences meeting. AIAA 2010-1318
- AIAA 3rd drag prediction workshop. URL: <http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop3/workshop3.html>, [Online]
- Andres E, Monge F (INTA), Perez A, Salcedo S (UAH) (2011) Metamodel assisted aerodynamic design using evolutionary optimization. EUROGEN
- Asouti VG, Giannakoglou KC (Technical University of Athens, Greece) (2009) A grid-27 asynchronous metamodel-assisted evolutionary algorithm for aerodynamic optimization. Genetic Programming and Evolvable Machines
- Bompard B, Peter J (ONERA, France), Desideri J (INRIA, France) (2010) Surrogate models based on function and derivative values for aerodynamic global optimization. ECOMAS CFD
- Branke J, Schmidt C (2005) Faster convergence by means of fitness estimation. Soft Comput Fusion Found Methodologies Appl 9(1):13–20

- Buche D, Schraudolph N, Koumoutsakos P (2005) Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Trans Syst Man Cybern Part C Appl Rev* 35:183–194
- Carrier G (2006) Single and multipoint aerodynamic optimizations of a supersonic transport aircraft wing using optimization strategies involving adjoint method and genetic algorithm. In: ERCOFTAC conference on optimization
- Cheng CS, Chen PW, Huang KK (2011) Estimating the shift size in the process mean with support vector regression and neural networks. *Expert Syst Appl* 38:10624–10630
- Clarke SM, Griebisch JH, Simpson TW (2005) Analysis of support vector regression for approximation of complex engineering analyses. *Trans ASME, J Mech Des* 127(6):1077–1087
- Cook P, McDoland M, Firmin M (1979) Aerofoil RAE 2822—pressure distributions, and boundary layer and wake measurements. AGARD Report 138
- de Weerd E, Chu QP, Mulder JA (Delft University of Technology, The Netherlands) (2005) Neural network aerodynamic model identification for aerospace reconfiguration. AIAA
- Duvigneau R, Visonneau M (2004) (Ecole Centrale de Nantes, France), Hybrid genetic algorithms and artificial neural networks for complex design optimization in CFD, *Int J Numer Methods Fluids* 44
- Epstein B, Jameson A, Peigin S, Roman D, Harrison N, Vassberg J (2008) Comparative study of 3D wing drag minimization by different optimization techniques. In: 46th AIAA aerospace sciences meeting and exhibit. AIAA paper 2, Reno, Nevada
- Forrester A, Keane A (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1–3):50–79
- Giannakoglou K (2002) Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Prog Aerosp Sci* 38(1):43–76
- Iuliano E, Andrés E (2015) “Application of surrogate-based global optimization to aerodynamic design.” Springer Tracts in Mechanical Engineering. ISBN 978-3-319-21505-1
- Iuliano E, Quagliarella D (2013) “Aerodynamic shape optimization via non-intrusive POD-based surrogate modelling.” In: IEEE congress on evolutionary computation. Cancún, México, June 20–23
- Jiang H, He W (2012) Grey relational grade in local support vector regression for financial time series prediction. *Expert Syst Appl* 39:2256–2262
- Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput J* 9(1):3–12
- Jin Y, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans Evol Comput* 6(5):481–494
- Kampolis IC, Giannakoglou KC (2011) Synergetic use of different evaluation, parametrisation and search tools within a hierarchical optimization platform. *Appl Soft Comput* 11(1):645–651
- Kapsoulis D, Tsiakas K, Asouti V, Giannakoglou K (2016) The use of Kernel PCA in evolutionary optimization for computationally demanding engineering applications. In: 2016 IEEE symposium series on computational intelligence (SSCI). Athens, Greece, pp 1–8. <https://doi.org/10.1109/ssci.2016.7850203>
- Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: IEEE international conference on neural networks
- Leifsson L, Koziel S, Tesfahunegn Y (2014) “Aerodynamic design optimization: physics-based surrogate approaches for airfoil and wing design”. AIAA SciTech. AIAA 2014-0572
- Lim D, Jin Y, Ong YS, Sendho B (2010) Generalizing surrogate-assisted evolutionary computation. *IEEE Trans Evol Comput* 14(3):329–355
- Marinus BG (Royal Military Academy/VKI for Fluid Dynamics, Belgium), Rogery M (Ecole Centrale de Lyon, France), Braembussch R (VKI for Fluid Dynamics, Belgium) (2010) Aeroacoustic and aerodynamic optimization of aircraft propeller blades. AIAA
- Muller SD, Marchetto J, Airaghi S, Koumoutsakos P (2002) Optimization based on bacterial chemotaxis. *IEEE Trans Evol Comput* 6(1):16–29
- Peter J, Marcelet M (ONERA, France) (2007) Comparison of surrogate models for turbomachinery design. In: 7th weas international conference on simulation, modelling and optimization

- Peter J, Carrier G, Bailly D, Klotz P, Marcelet M, Renac F (March 2011) Local and Global Search Methods for Design in Aeronautics. ONERA J AerospaceLab 2
- Praveen C, Duvigneau R (2007) Radial basis functions and kriging metamodels for aerodynamic optimization. INRIA Technical report
- Salcedo-Sanz S, Ortiz-Garcia E, Perez-Bellido A, Portilla A (2011) Short term wind speed prediction based on evolutionary support vector regression algorithms. *Expert Syst Appl* 38:4052–4057
- Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11:341–359
- Szollós A, Smid M, Hájek J (2009) Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation: knowledge-based reinitialization, crowding and dominance. *Adv Eng Software* 40
- Vankan J, Maas R (2010) Surrogate modeling for efficient design optimization of composite aircraft fuselage panels. In: 27th international congress of the aeronautical sciences, ICAS
- Won KS, Ray T (2005) A Framework for Design Optimization using Surrogates. *Eng Optim* 37(7):685–703
- Younis A, Jichao G, Zuomin D, Guangyao L (2008) Trends, features, and test of common and recently introduced global optimization methods. In: 2th AIAA/ISSMO multidisciplinary & optimization conference. AIAA 2008-5853
- Zhong-Hua H, Zimmermann R, Görtz S (DLR, Germany) (2010) A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data. AIAA

A Response Surface Based Strategy for Accelerated Compressor Map Computation



Dmitrij Ivanov, Dieter Bestle and Christian Janke

Abstract The aim of the present work is to develop a strategy enabling fast CFD-based computation of compressor maps for aero engines. The introduced process consists of two phases. In the first phase the compressor limits due to surge and choke are identified and approximated by utilizing methods of support vector machine (SVM). These limit lines are refined within an iterative, distance-based approach. Subsequently, in the second phase the three-dimensional shape of the compressor map is approximated by a response surface method (RSM). The process is validated with an application to an industrial 4.5-stage research compressor, where very good agreement between evaluated and approximated values is obtained.

Introduction

The rapid development of computer power, design methods and process integration allows to switch step-by-step from manually driven to automated processes and from low-fidelity to high-fidelity analysis. This is especially the case in aero engine design, Keskin (2007), and compressor map computation, Janke et al. (2015), as last step of the design process.

For a long time engine experts have been denying that compressor map computation based on 3D-CFD would be possible without human intelligence due to the complexity of detecting surge- and choke-limits and the high sensitivity of CFD

D. Ivanov (✉) · D. Bestle

Engineering Mechanics and Vehicle Dynamics, Brandenburg University of Technology
Cottbus-Senftenberg, Siemens-Halske-Ring 14, 03046 Cottbus, Germany
e-mail: ivanov@b-tu.de

D. Bestle

e-mail: bestle@b-tu.de

C. Janke

Rolls-Royce Deutschland Ltd & Co KG,
Eschenweg 11, 15827 Blankenfelde-Mahlow, Dahlewitz, Germany
e-mail: Christian.Janke@Rolls-Royce.com

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_14

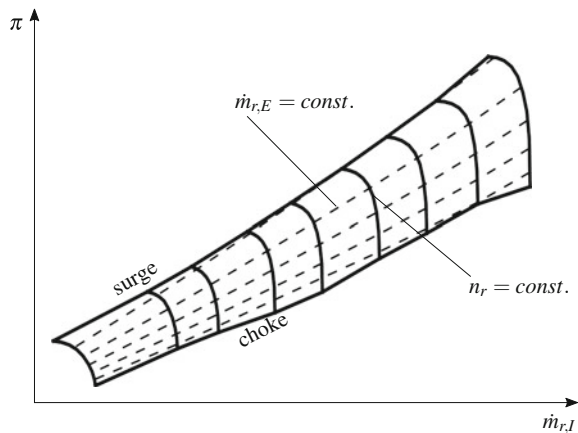
convergence behavior on initial flow conditions. In common industrial applications, therefore, typically a more robust 1D-CFD analysis, a so called meanline-code, is used to calculate compressor maps. This code calculates the flow along the mid-line between hub and shroud and captures the influence of three-dimensional flow phenomena by correlations only, Keskin (2007).

However, one of the authors, Janke et al. (2015), could demonstrate for an industrial 4.5-stage compressor test rig that integration of optimization and root search algorithms for critical flow conditions can resolve this problem. Basically this approach follows the classical root of computing so called speed-lines for constant reduced engine speeds $n_r = const$, see Fig. 1. By changing the reduced exit mass flow $\dot{m}_{r,E}$ it moves along these speed-lines and finds the limits due to surge and choke. Although this approach was able to compute a high-fidelity compressor map based on the 3D-CFD inhouse code HYDRA, Lapworth (2004), without any human interaction, the computational time of more than one week is still too high for industrial application.

Therefore, the presented approach in this paper will be released from following speed-lines and use more general design of experiments (DoE) strategies like Latin hypercube sampling (LHS). It will combine response surface methods (RSM) and support vector machine (SVM). Statistical learning methods like artificial neuronal networks, Fei et al. (2016); Ghorbanian and Gholamrezaei (2009) or SVM Sodemann et al. (2006), have shown to be beneficial for data-driven compressor map modelling, however this was based on a sufficient big set of training points, whereas the presented approach starts with a small set of points and explores the design space autonomously. To the authors' knowledge, this kind of approach has not been applied to compressor map computation before.

The paper is organized as follows: After a short introduction into the physical background of compressor map physics, the two-phase procedure for automatically computing the map is introduced. In the first phase, the surge- and choke-limit lines are successively determined by an iterative refinement procedure utilizing SVM

Fig. 1 Classical representation of a compressor map as function of inlet mass flow, Janke et al. (2015)



with kernel trick based on a nonlinear transformation with radial basis functions. Thereafter, the three-dimensional shapes of the compressor map functions are approximated with RSM. Finally the whole process is validated by computing a compressor map of an industrial 4.5-stage research compressor, Janke et al. (2015). Since the focus of the paper is on development of a new strategy for computing compressor maps, the meanline code is used instead of HYDRA whenever CFD computations are mentioned in this paper. This, however, is no real restriction, actual investigations of coupling the proposed approach with 3D-CFD analysis by the industrial code HYDRA show promising results. The entire process is embedded into a framework written in PYTHON.

Theoretical Background of Compressor Map Computation

The compressor map of an aero engine shows the ratio

$$\pi = \frac{p_{t,E}}{p_{t,I}} \quad (1)$$

between total pressures at the compressor inlet $p_{t,I}$ and outlet $p_{t,E}$ and the efficiency

$$\eta = \frac{\pi^{\frac{\kappa-1}{\kappa}} - 1}{\frac{T_{t,E}}{T_{t,I}} - 1} \quad (2)$$

as functions of reduced compressor inlet massflow

$$\dot{m}_{r,I} = \frac{\dot{m}_I \sqrt{T_{t,I}}}{p_{t,I}} \quad (3)$$

and reduced shaft speed

$$n_r = \frac{n}{\sqrt{T_{t,I}}}, \quad (4)$$

where κ denotes the isentropic exponent, \dot{m}_I the inlet mass flow, n the shaft speed and $T_{t,\bullet}$ the total temperature. A classical representation of the pressure ratio π is shown in Fig. 1, where additionally lines $\dot{m}_{r,E} = \text{const.}$ are visualized. The reason is that for CFD and rig-testing the reduced compressor exit massflow

$$\dot{m}_{r,E} = \frac{\dot{m}_E \sqrt{T_{t,E}}}{p_{t,E}} \quad (5)$$

is used as an independent variable instead of the reduced inlet massflow.

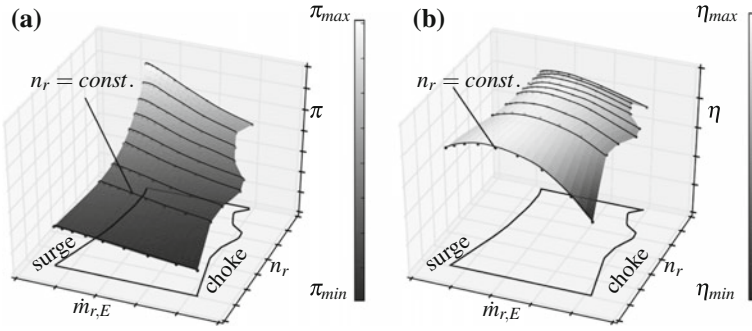


Fig. 2 Three-dimensional extension of compressor map for **a** pressure ratio and **b** efficiency

Despite the fact that compressor characteristics are two-dimensional functions of $\dot{m}_{r,I}$ and n_r , in common industrial applications compressor maps are represented only as one-dimensional graphs where the pressure ratio $\pi(\dot{m}_{r,I})$ is calculated along user-defined speed-lines $n_r = const.$, Fig. 1. Every speed-line is limited by the compressor operation limits surge and choke and the connection lines of all surge- and choke-points are called surge- and choke-line, respectively. The knowledge of these limiting lines is crucial for engine performance, reliability and safety assessment during flight. If the compressor operates on a state between the evaluated speed-lines, typically a linear interpolation is performed. This makes clear that, depending on the number of calculated speed-lines, the accuracy of prediction of compressor characteristics for any arbitrary operation point rises. The work presented here tries to catch the full three-dimensional character of a compressor map. Similar to rig-testing, instead of the reduced inlet massflow $\dot{m}_{r,I}$ the reduced exit massflow $\dot{m}_{r,E}$ and n_r are used as independent variables to describe the operating state resulting in $\mathbf{x} = [\dot{m}_{r,E}, n_r]^T$ and

$$\pi = \pi(\mathbf{x}), \eta = \eta(\mathbf{x}), \dot{m}_{r,I} = \dot{m}_{r,I}(\mathbf{x}), \tag{6}$$

respectively. Three-dimensional representations of $\pi(\mathbf{x})$ and $\eta(\mathbf{x})$, resulting from meanline analysis of the already mentioned 4.5-stage research compressor investigated by Janke et al. (2015) and shown in Fig. 2, demonstrate that they are smooth enough to be approximated by response surfaces. In order to obtain these functions, firstly the limits regarding choke and surge are determined and then the map characteristics in between are approximated.

Separation of Classified Points with Support Vector Machines

Surge- and choke-lines are separation curves between sample points \mathbf{x}_i in the $(\dot{m}_{r,E}, n_r)$ -plane where CFD is converging on one side, and for points on the other side the flow calculation cannot be performed either because of choke or surge,

respectively. This may be classified as

$$y_i = \begin{cases} +1 & \text{if CFD converges,} \\ -1 & \text{else.} \end{cases} \quad (7)$$

Thus, the goal is to find a classification border separating the data

$$\{(\mathbf{x}_i, y_i) \mid i = 1(1)N, y_i \in \{-1, 1\}\} \quad (8)$$

according to their y -value.

Let us firstly assume that samples \mathbf{x}_i are linearly separable by a line H given by a normal vector $\mathbf{w} \perp H$ and a point $\mathbf{x}_0 \in H$ on the line, i.e.,

$$(\mathbf{x} - \mathbf{x}_0) \perp \mathbf{w} \quad \text{or} \quad \mathbf{w}^T(\mathbf{x} - \mathbf{x}_0) = 0 \quad \forall \mathbf{x} \in H. \quad (9)$$

With $b := -\mathbf{w}^T \mathbf{x}_0$ the separation line is given as

$$H : \mathbf{w}^T \mathbf{x} + b = 0, \quad b \in \mathbb{R}, \quad (10)$$

where \mathbf{w} and b have to be selected such that

$$\mathbf{w}^T \mathbf{x}_i + b = \begin{cases} > 0 & \text{for } y_i = +1, \\ < 0 & \text{for } y_i = -1. \end{cases} \quad (11)$$

From Fig. 3 it is obvious that there is no unique solution for such a separation line. All dashed and solid lines in the figure fulfill condition (11). Finding the best separation line, i.e. best \mathbf{w} and b , may be solved with the concept of support vector machine.

For later use, the conditions (11) may be combined in single inequality constraints

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 0, \quad i = 1(1)N, \quad (12)$$

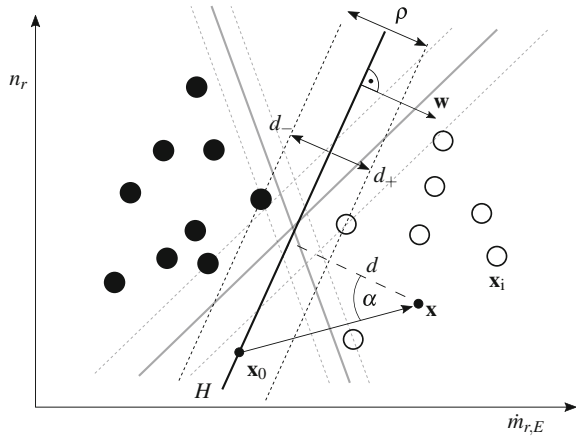
respectively. According to Fig. 3, the signed distance of any point \mathbf{x} to H is given as

$$\begin{aligned} d &= \|\mathbf{x} - \mathbf{x}_0\| \cos(\alpha) \equiv \frac{\|\mathbf{w}\| \|\mathbf{x} - \mathbf{x}_0\| \cos(\alpha)}{\|\mathbf{w}\|} \\ &= \frac{\mathbf{w}^T(\mathbf{x} - \mathbf{x}_0)}{\|\mathbf{w}\|} = \frac{\mathbf{w}^T \mathbf{x} - \mathbf{w}^T \mathbf{x}_0}{\|\mathbf{w}\|} \end{aligned} \quad (13)$$

or with $b = -\mathbf{w}^T \mathbf{x}_0$

$$d = \frac{\mathbf{w}^T \mathbf{x} + b}{\|\mathbf{w}\|} \quad (14)$$

Fig. 3 Example of separation of feasible (○) and infeasible (●) samples by linear decision manifold



where $\|\cdot\|$ notes the Euclidean norm. The classification of a new sample point $\mathbf{x} \notin H$ is then given as

$$y = \text{sign}(d) \equiv \text{sign}(\mathbf{w}^T \mathbf{x} + b) \in \{-1, 1\}. \tag{15}$$

With Eq. 14, the geometric margin in Fig. 3 is given as

$$\rho = \underbrace{\min_{i: y_i=1} \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|}}_{d_+} + \underbrace{\min_{i: y_i=-1} \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|}}_{d_-}. \tag{16}$$

For an optimal separation, the value of b may be defined such that $d_+ = d_-$ and the normal vector \mathbf{w} may be normalized such that according to Vapnik (1999)

$$\min_{i: y_i=1} |\mathbf{w}^T \mathbf{x}_i + b| = \min_{i: y_i=-1} |\mathbf{w}^T \mathbf{x}_i + b| \stackrel{!}{=} 1. \tag{17}$$

Hence, the geometric margin in Eq. (16) becomes

$$\rho = \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}, \tag{18}$$

and with Eq. (17) the inequality condition (12) may be changed to

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad \forall \mathbf{x}_i. \tag{19}$$

Even with these restrictions, the separation line is still not unique, see different solid separation lines with dashed geometric margins in Fig. 3. It becomes unique by introducing an optimality criterion. According to Vapnik (1999) a dataset (8) is optimally separated by H if the geometric margin becomes largest. This results in

$$\max \rho = \max \frac{2}{\|\mathbf{w}\|} \equiv \min \|\mathbf{w}\| \equiv \min_{\mathbf{w}, b} \mathbf{w}^T \mathbf{w} \text{ s.t. Eq. (19).} \quad (20)$$

The constrained optimization problem may be solved by using Karush-Kuhn-Tucker conditions with the Lagrange function

$$L(\mathbf{w}, b, \boldsymbol{\lambda}) = \mathbf{w}^T \mathbf{w} - \sum_{i=1}^N \lambda_i [1 - y_i (\mathbf{w}^T \mathbf{x}_i + b)] \quad (21)$$

and Lagrange multipliers $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_N]$, $\lambda_i \leq 0$. According to the appendix, the primal problem (20) can be transformed into a dual problem, Schoelkopf (2001); Vapnik (1999); Vapnik et al. (1995),

$$\boldsymbol{\lambda}^* = \max_{\boldsymbol{\lambda} \leq 0} L(\boldsymbol{\lambda}) \text{ s.t. } 0 = \sum_{i=1}^N \lambda_i y_i \quad (22)$$

with

$$L(\boldsymbol{\lambda}) = -\frac{1}{4} \sum_{j=1}^N \sum_{i=1}^N \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^N \lambda_i. \quad (23)$$

This quadratic problem can be solved very efficiently. Based on Eq. (61) the optimal normal vector is given by

$$\mathbf{w}^* = -\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i. \quad (24)$$

Let us choose any point $\mathbf{x}^+ \in \{\mathbf{x}_i | y_i = +1, \lambda_i^* \neq 0\}$ and any point $\mathbf{x}^- \in \{\mathbf{x}_i | y_i = -1, \lambda_i^* \neq 0\}$ where constraints (19) are active. The offset b^* may then be determined from Eq. (64) as

$$b^* = -\frac{1}{2} \left(\mathbf{w}^{*T} \mathbf{x}^+ + \mathbf{w}^{*T} \mathbf{x}^- \right). \quad (25)$$

With these optimal solutions the classification (15) for any sample point \mathbf{x} reads as

$$\begin{aligned} y(\mathbf{x}) &= \text{sign} \left(\mathbf{w}^{*T} \mathbf{x} + b^* \right) \\ &= \text{sign} \left(-\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i^T \mathbf{x} \right. \\ &\quad \left. - \frac{1}{2} \sum_{i=1}^N \lambda_i y_i \mathbf{x}_i^T \mathbf{x}^+ - \frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \mathbf{x}_i^T \mathbf{x}^- \right). \end{aligned} \quad (26)$$

In general, surge- and choke-limits are not straight, but curved lines, and thus not linearly separable. In this case, the kernel trick may be used where the data \mathbf{x}_i are transformed into a higher dimension $D > 2$ by

$$\tilde{\mathbf{x}}_i = \boldsymbol{\varphi}(\mathbf{x}_i), \quad \boldsymbol{\varphi} : \mathbb{R}^2 \rightarrow \mathbb{R}^D, \quad i = 1(1)N. \quad (27)$$

If the dimension D is high enough, the data $\tilde{\mathbf{x}}_i$ become linearly separable, Schoelkopf (2001), Vapnik (1999), Vapnik et al. (1995). Thus, there exists an optimal, separating hyperplane (10) in the $\tilde{\mathbf{x}}$ -space where

$$\boldsymbol{\lambda}^* = \max_{\boldsymbol{\lambda} \leq 0} L(\boldsymbol{\lambda}) \quad \text{s.t.} \quad 0 = \sum_{i=1}^N \lambda_i y_i \quad (28)$$

with

$$L(\boldsymbol{\lambda}) = -\frac{1}{4} \sum_{j=1}^N \sum_{i=1}^N \lambda_i \lambda_j y_i y_j \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j - \sum_{i=1}^N \lambda_i \quad (29)$$

and

$$\tilde{\mathbf{w}}^* = -\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i \tilde{\mathbf{x}}_i, \quad \tilde{b}^* = -\frac{1}{2} \left(\tilde{\mathbf{w}}^{*T} \tilde{\mathbf{x}}^+ + \tilde{\mathbf{w}}^{*T} \tilde{\mathbf{x}}^- \right) \quad (30)$$

which may be easily obtained by substituting all associated quantities \bullet as $\tilde{\bullet}$ in Eqs. (23)–(25). For classification, the transformation function (27) itself is not required, but only dot products

$$K(\mathbf{x}_i, \mathbf{x}_j) := \tilde{\mathbf{x}}_i^T \tilde{\mathbf{x}}_j = \boldsymbol{\varphi}^T(\mathbf{x}_i) \boldsymbol{\varphi}(\mathbf{x}_j). \quad (31)$$

This is obvious in Eq. (29) for finding $\boldsymbol{\lambda}^*$ and can also be seen in the classification operator which analogous to Eq. (26) reads as

$$y(\mathbf{x}) = \text{sign} \left(-\frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}) - \frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}^+) - \frac{1}{2} \sum_{i=1}^N \lambda_i^* y_i K(\mathbf{x}_i, \mathbf{x}^-) \right). \quad (32)$$

The dot products (31) are called kernel function, and here we will use the bell-shaped radial basis function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp \left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma} \right) \quad (33)$$

where in numerical studies $\sigma = 0.1$ was found to perform well. For the SVM-related tasks the PYTHON code package scikit-learn is used, Pedregosa et al. (2011).

Estimation of Choke and Surge Limits

Let us now apply the concept to the surge line

$$\dot{m}_{r,E} = f_S(n_r) \quad (34)$$

to be found. Based on the design point $\mathbf{x}^{DP} = [\dot{m}_{r,E}^{DP}, n_r^{DP}]^T$ and minimum and maximum reduced shaft speeds we choose a rectangular search region

$$[\dot{m}_{r,E}^{DP} - \Delta\dot{m}_r, \dot{m}_{r,E}^{DP}] \times [n_{r,min}, n_{r,max}] \quad (35)$$

with prescribed width $\Delta\dot{m}_r$. The starting set of training points (8) is chosen to consist of the four corner points, the mid-point and five randomly chosen points, see Fig. 4a ($\nu = 1$). They are evaluated with the CFD analysis and according to the outcome and Eq. (7) they are classified either as converged by $y_i = 1$ (white dots) or non-converged by $y_i = -1$ (black dots). Based on this, the separating hyperplane in the $\tilde{\mathbf{x}}$ -space may be characterized by Eq. (30).

However, this is only theoretical, since the transformation function (27) is not known. In order to describe the separation line in the original \mathbf{x} -space, the search space (35) is divided by a fine rectangular grid of $N_g \times N_g$ points as

$$\mathbf{x}_{i,j} = [\dot{m}_{r,E,j}, n_{r,i}]^T, \quad i = 1(1)N_g, \quad j = 1(1)N_g, \quad (36)$$

where here $N_g = 300$ is chosen. These points are then very fast characterized by applying Eq. (32) resulting in classification numbers $y_{i,j}$. For each shaft speed $n_{r,i}$ the corresponding point $f_S(n_{r,i})$ of surge line (34) is found approximately as

$$f_S(n_{r,i}) = \frac{\dot{m}_{r,E,k} + \dot{m}_{r,E,k+1}}{2} \quad \text{where } k : y_{i,k} y_{i,k+1} < 0, \quad i = 1(1)N_g. \quad (37)$$

The black line in Fig. 4a ($\nu = 1$) shows these $N_g = 300$ limit points with a linear interpolation in between.

In order to find the real surge line (dashed line in Fig. 4a, $\nu = 1$), the grid points (36) may be evaluated directly by CFD and classified correctly by Eq. (7). The line is then found also according to Eq. (37). Obviously the surge line found by SVM deviates very much from this real surge limit. The reason is that the SVM result is based on an insufficient number of support points which, therefore, should be updated iteratively.

Let us summarize the support points (8) used so far in a set

$$X^{(1)} = \left\{ \mathbf{x}_i = [\dot{m}_{r,E,i}, n_{r,i}]^T \mid i = 1(1)10 \right\} \quad (38)$$

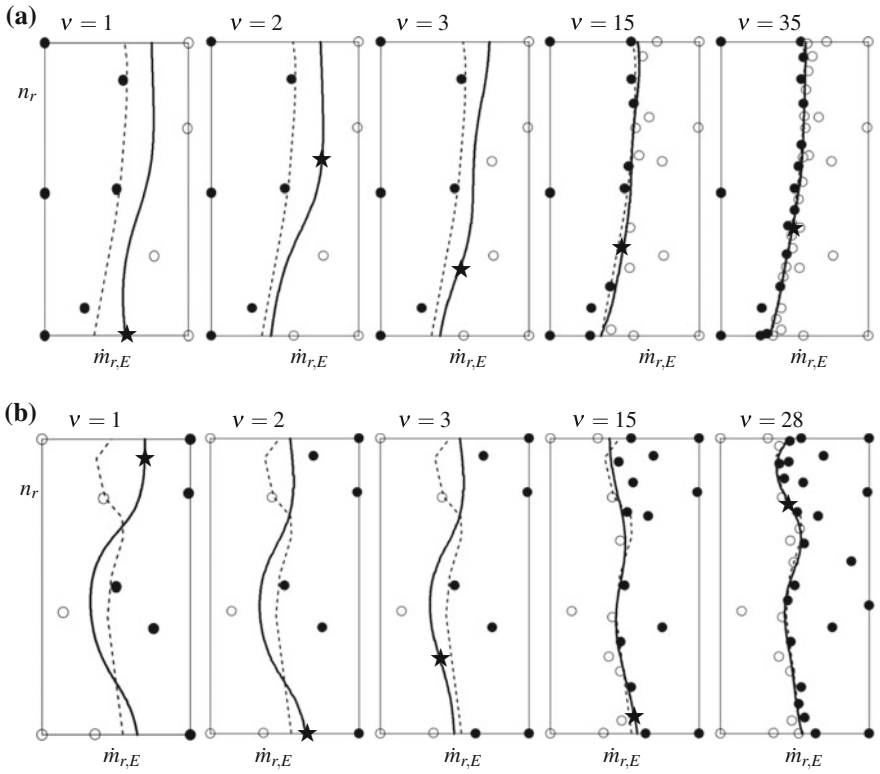


Fig. 4 Iterative refinement of **a** surge- and **b** choke-line based on converged (\circ) and diverged (\bullet) CFD solutions and chosen point for refinement (\star) with approximated (solid) and real (dashed) limit curves

and the found limit points (37) of our first approximation $\dot{m}_{r,E} = f_S^{(1)}(n_r)$ in a set

$$\bar{X}^{(1)} = \left\{ \bar{\mathbf{x}}_i = [f_S(n_{r,i}), n_{r,i}]^T \mid i = 1(1)N_g \right\}. \tag{39}$$

Then each point $\bar{\mathbf{x}}_i \in \bar{X}^{(1)}$ is a potential candidate to adequately update $X^{(1)}$. The most can be gained by using the limit point with the largest normalized distance from all support points used so far, i.e.,

$$\begin{aligned}
 X^{(2)} &:= X^{(1)} \cup \left\{ \bar{\mathbf{x}}_k \in \bar{X}^{(1)} \right\} \\
 &\text{where } k = \arg \max_i d(\bar{\mathbf{x}}_i, X^{(1)}), \\
 d(\bar{\mathbf{x}}, X) &= \min_{\mathbf{x}_j \in X} \|\bar{\mathbf{x}} - \mathbf{x}_j\|.
 \end{aligned}
 \tag{40}$$

The norm used here is applied to normalized variables, i.e.,

$$\|\Delta \mathbf{x}\| = \sqrt{\left(\frac{\Delta \dot{m}_{r,E}}{\dot{m}_{r,E,max}}\right)^2 + \left(\frac{\Delta n_r}{n_{r,max}}\right)^2}. \quad (41)$$

In Fig. 4a ($\nu = 1$) the update point is marked by a black star and the resulting limit curve $f_S^{(2)}(n_r)$ in Fig. 4a ($\nu = 2$) partly comes closer to the real surge line. It should be mentioned that the scales of abscissa $\dot{m}_{r,E}$ and ordinate n_r in Fig. 4 differ in magnitude. The magnitude of massflow is $10^1 \leq \dot{m}_{r,E} \leq 10^2$ and that of reduced shaft speed $10^2 \leq n_r \leq 10^3$, which is why the largest distance between points cannot be directly seen in Fig. 4.

This update procedure $X^{(\nu)} \rightarrow X^{(\nu+1)}$ may be continued until the root mean square (RMS) of changes in the limit line

$$RMS = \sqrt{\frac{1}{N_g} \sum_{k=1}^{N_g} \left(f_S^{(\nu)}(n_{r,k}) - f_S^{(\nu-1)}(n_{r,k})\right)^2} \leq \varepsilon \quad (42)$$

falls below a tolerance, e.g. $\varepsilon = 0.05$. As depicted in Fig. 4a, this happens after 34 updates with a satisfying result $f_S^{(35)}(n_r) \approx f_S(n_r)$.

The same procedure is applied to the choke region

$$[\dot{m}_{r,E}^{DP}, \dot{m}_{r,E}^{DP} + \Delta \dot{m}_r] \times [n_{r,min}, n_{r,max}] \quad (43)$$

resulting in an even better result for the choke-line

$$\dot{m}_{r,E} = f_C(n_r) \quad (44)$$

after 27 iterations.

RSM Approximations of Compressor Maps

For the approximation of the compressor map characteristics (6) all converged points (\circ) in Fig. 4 during surge- and choke-line computation may be used. However, the space between the two limit lines (34) and (44) has a very low point density. Thus, the space between surge- and choke-line must be seeded with additional points.

For this purpose, a Latin hypercube sampling (LHS) approach with N_s points with equal density on a unit square

$$\mathbf{u}_i = [\xi_i, \eta_i]^T \in [0, 1]^2, \quad i = 1(1)N_s, \quad (45)$$

is performed. Then these samples are transformed into the admissible compressor map range by

$$n_{r,i} = n_{r,min} + (n_{r,max} - n_{r,min}) \eta_i, \tag{46}$$

$$\dot{m}_{r,E,i} = f_S(n_{r,i}) + (f_C(n_{r,i}) - f_S(n_{r,i})) \xi_i \tag{47}$$

based on (34) and (44), see Fig. 5.

The training points for surge- and choke-line approximation (o) and new points from LHS (+) are concatenated to one dataset $\{\mathbf{x}_i\}$ (see Fig. 6) to create a RSM model $\hat{\pi}(\mathbf{x})$ by combining a linear regression function and radial basis functions under tension, Bouhamidi and Méhauté (2004):

$$\hat{\pi}(\mathbf{x}) = b_0 + [b_1, b_2] \mathbf{x} + \sum_{i=1}^N a_i \psi(\|\mathbf{x} - \mathbf{x}_i\|), \quad N \geq 3, \tag{48}$$

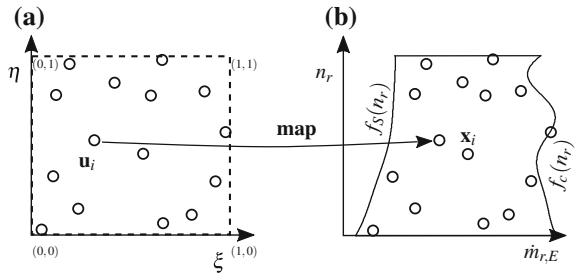
where

$$\psi(\|\Delta \mathbf{x}\|) = \|\Delta \mathbf{x}\| \ln \|\Delta \mathbf{x}\|. \tag{49}$$

The coefficients a_i, b_0, b_1, b_2 are found from interpolation conditions

$$\hat{\pi}(\mathbf{x}_i) = \pi(\mathbf{x}_i) \quad \forall \mathbf{x}_i \tag{50}$$

Fig. 5 Transformation of LHS samples from **a** unit square into **b** admissible compressor map region



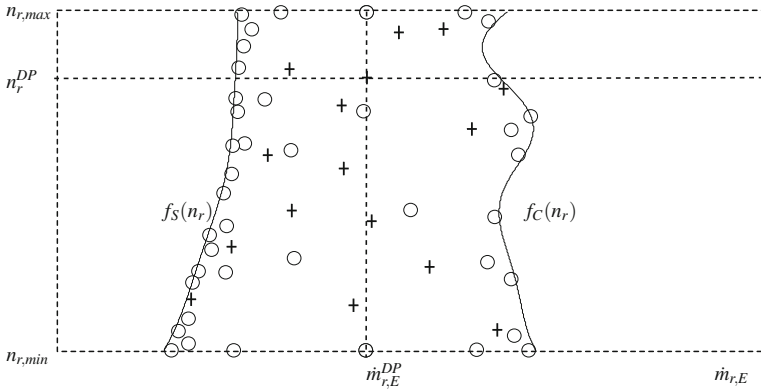


Fig. 6 Samples for compressor map approximation summarizing converged points (○) of surge- and choke-line (solid lines) computation and 16 additional points (+) from LHS sampling

and

$$\sum_{i=1}^N a_i = 0, \quad \sum_{i=1}^N a_i \mathbf{x}_i = \mathbf{0}. \tag{51}$$

Analogously response surfaces $\hat{\eta}(\mathbf{x})$ and $\hat{m}_{r,I}(\mathbf{x})$ are created for efficiency and reduced inlet massflow.

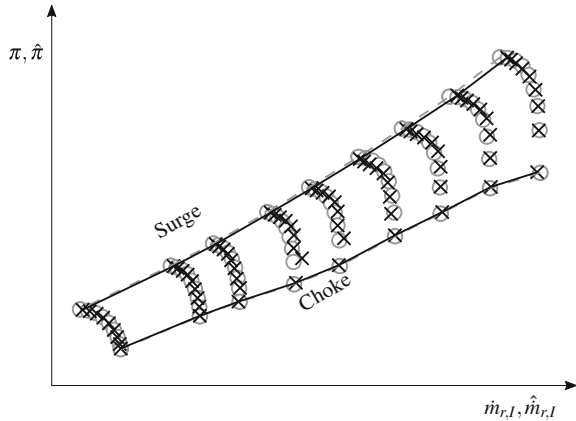
Finally the process is validated by computing compressor map characteristics of the 4.5-stage high-speed research compressor, Janke et al. (2015), and comparing them with a fully evaluated dataset. In principal, a detailed comparison of RSM models for $\hat{m}_{r,I}(\hat{m}_{r,E}, n_r)$, $\hat{\eta}(\hat{m}_{r,E}, n_r)$ and $\hat{\pi}(\hat{m}_{r,E}, n_r)$ should be illustrated. But to align to the classical representation of a compressor map, only a one-dimensional representation of $\hat{\pi}(\hat{m}_{r,I})$ for several speed-lines $n_r = const.$ is chosen.

Nine $\hat{m}_{r,E}$ -values and the associated $\hat{m}_{r,I}$ -values for nine speed-lines are extracted from a directly evaluated dataset. The $\hat{m}_{r,E}$ -values from the dataset are also used to calculate the approximated values $\hat{m}_{r,I}$ and $\hat{\pi}$ from the response surface described above. From Fig. 7 a good approximation can be observed. The representation in Fig. 7 is suitable to easily identify approximation errors, where horizontal misalignment indicates approximation errors in $\hat{m}_{r,I}(\mathbf{x})$ and vertical misalignment indicates approximation errors in $\hat{\pi}(\mathbf{x})$.

In order to check approximation quality, the entire compressor map calculation process was performed 50 times. For each run maximum relative approximation errors

$$e_{\pi} = \max \left| \frac{\hat{\pi}(\mathbf{x}) - \pi(\mathbf{x})}{\pi(\mathbf{x})} \right|, \quad e_{\hat{m}_{r,I}} = \max \left| \frac{\hat{m}_{r,i}(\mathbf{x}) - \dot{m}_{r,i}(\mathbf{x})}{\dot{m}_{r,i}(\mathbf{x})} \right|, \quad e_{\eta} = \max \left| \frac{\hat{\eta}(\mathbf{x}) - \eta(\mathbf{x})}{\eta(\mathbf{x})} \right| \tag{52}$$

Fig. 7 Comparison between RSM (×) and directly evaluated (○) map data as well as evaluated (dashed) and SVM based (solid) surge- and choke-lines



were captured. Additionally displacements between evaluated and approximated surge- and choke-line positions

$$e_S = \max \left| \frac{f_S(n_{r,j}) - \dot{m}_{r,E}^S(n_{r,j})}{\dot{m}_{r,E}^S(n_{r,j})} \right|, \quad e_C = \max \left| \frac{f_C(n_{r,j}) - \dot{m}_{r,E}^C(n_{r,j})}{\dot{m}_{r,E}^C(n_{r,j})} \right|, \quad j = 1(1)9, \tag{53}$$

for all nine speed-lines were determined. Statistical evaluation led to the following mean approximation errors:

$$\bar{e}_\pi = 5.7\%, \quad \bar{e}_{\dot{m}_{r,l}} = 3.2\%, \quad \bar{e}_\eta = 17.2\%, \quad \bar{e}_C = 2.1\%, \quad \bar{e}_S < 1\%. \tag{54}$$

Obviously choke- and surge-lines are predicted very well. The higher approximation errors in between are high due to the low number of support points. It could be easily reduced by using larger number of samples, however, in view of application of the process strategy to 3D-CFD the number of evaluations was kept at a minimum.

Conclusions and Outlook

A novel approach for fast compressor map computation based on SVM and RSM is introduced. It turns out that approximating surge- and choke-lines based on SVM and iterative curve refinement works very well. Also the approximation of compressor characteristics by RSM could be validated. Due to the use of a 1D-CFD code, detailed statistical investigations of the proposed strategy were performed. However, the procedure now has to be validated with 3D-CFD.

Acknowledgements This work has been carried out in collaboration with Rolls-Royce Deutschland as part of the research project VITIV (Virtual Turbomachinery, Proj.-No. 80164702) funded by the State of Brandenburg, the European Regional Development Fund, and Rolls-Royce Deutschland. Rolls-Royce Deutschland's permission to publish this work is greatly acknowledged.

Appendix

In order to solve the optimization problem (20), matrices

$$\mathbf{y} = [y_1 \dots y_N]^T, \mathbf{1} = [1 \dots 1]^T, \mathbf{X} = [y_1 \mathbf{x}_1 \dots y_N \mathbf{x}_N] \quad (55)$$

are introduced. The Lagrange function (21) then simplifies to

$$L = \mathbf{w}^T \mathbf{w} - \boldsymbol{\lambda}^T (\mathbf{1} - \mathbf{X}^T \mathbf{w} - b \mathbf{y}) \quad (56)$$

and the Karush-Kuhn-Tucker conditions, Bestle (1994), read as

$$\frac{\partial L}{\partial \mathbf{w}} = 2\mathbf{w} + \mathbf{X}\boldsymbol{\lambda} = \mathbf{0}, \quad (57)$$

$$\frac{\partial L}{\partial b} = \boldsymbol{\lambda}^T \mathbf{y} = 0, \quad (58)$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} = -(\mathbf{1} - \mathbf{X}^T \mathbf{w} - b \mathbf{y}) \geq \mathbf{0}, \quad (59)$$

$$\boldsymbol{\lambda} \leq \mathbf{0}, \lambda_i [1 - y_i \mathbf{x}_i^T \mathbf{w} - b y_i] = 0 \quad \forall i. \quad (60)$$

Substituting

$$\mathbf{w} = -\frac{1}{2} \mathbf{X}\boldsymbol{\lambda} \quad (61)$$

from (57) and inserting it into (56) simplifies the Lagrange function to

$$L(\boldsymbol{\lambda}) = \frac{1}{4} \boldsymbol{\lambda}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \mathbf{1} - \frac{1}{2} \boldsymbol{\lambda}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\lambda} \equiv -\frac{1}{4} \boldsymbol{\lambda}^T \mathbf{X}^T \mathbf{X} \boldsymbol{\lambda} - \boldsymbol{\lambda}^T \mathbf{1} \quad (62)$$

or Eq. (23), where the term $b\boldsymbol{\lambda}^T \mathbf{y}$ vanishes due to Eq. (58). The dual problem summarizes maximization of the Lagrange function w.r.t. $\boldsymbol{\lambda}$, the condition $\boldsymbol{\lambda} \leq \mathbf{0}$ from (60) and constraint (58). Substituting the optimal solution (22) in (61) results in the optimal normal vector (24). The optimal offset may be found from any of the active conditions, where $\lambda_i < 0$ in (60) results in

$$b^* = \frac{1}{y_i} - \mathbf{x}_i^T \mathbf{w}^* \equiv y_i - \mathbf{x}_i^T \mathbf{w}^* \quad (63)$$

due to $y_i = \pm 1$. The mean value of all N_a active constraints is then

$$b^* = \frac{1}{N_a} \left(\sum_{i=1}^{N_a} y_i - \sum_{i=1}^{N_a} \mathbf{x}_i^T \mathbf{w}^* \right). \quad (64)$$

Alternatively only $N_a = 2$ active training points \mathbf{x}^+ , \mathbf{x}^- on opposite sides of the separation line with $y^+ = +1$ and $y^- = -1$ may be used canceling the first sum in (64) and yielding Eq. (25), Vapnik (1999).

References

- Bestle D (1994) *Analyse und Optimierung von Mehrkörpersystemen*. Springer, Berlin
- Bouhamidi A, Le Méhauté A (2004) Radial basis functions under tension. *J Approximation Theory* 127:135–154. <https://doi.org/10.1016/j.jat.2004.03.005>
- Fei J, Zhao N, Shi Y, Feng Y, Wang Z (2016) Compressor performance prediction using a novel feed-forward neural network based on Gaussian kernel function. *Adv Mech Eng* 8. <https://doi.org/10.1177/1687814016628396>
- Ghorbanian K, Gholamrezaei M (2009) An artificial neural network approach to compressor performance prediction. *Appl Energy* 86:1210–1221. <https://doi.org/10.1016/j.apenergy.2008.06.006>
- Janke C, Bestle D, Becker B (2015) Compressor map computation based on 3D CFD analysis. *CEAS Aeronaut J* 6:515–527
- Keskin A (2007) *Process integration and automated multi-objective optimization supporting aerodynamic compressor design*, Dissertation, BTU Cottbus. Shaker, Aachen
- Lapworth B (2004) HYDRA-CFD: A framework for collaborative CFD development. In: *International conference on scientific and engineering computation*. IC-SEC, Singapore
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12:2825–2830
- Schoelkopf B (2001) *Learning with kernels*. The MIT Press, Cambridge
- Sodemann A, Li Y, Lee J, Lancaster R (2006) Data-driven surge map modeling for centrifugal air compressors. In: *ASME international mechanical engineering congress and exposition*. ASME, Chicago
- Vapnik V (1999) *The nature of statistical learning theory*. Springer, New York
- Vapnik V, Cortes C, Saitta L (1995) *Support-vector networks*. In: *Machine learning*. Kluwer Academic Publishers, Boston

Surrogate-Based Shape Optimization of the ERCOFTAC Centrifugal Pump Impeller



Remo De Donno, Stefano Rebay and Antonio Ghidoni

Abstract Centrifugal pumps are largely used in several fields and for different applications. Despite their wide diffusion, they are often not optimized for working at the design conditions. The aim of this paper is to investigate the potentialities offered by surrogate-based optimization techniques in centrifugal pump impeller shape optimization, to obtain a robust and fast algorithm for performance improvement. The geometry chosen for validating the proposed method is the ERCOFTAC centrifugal pump where accurate measurements and simulations are available in the literature. The three-dimensional geometry of the impeller is parametrized by means of parametric Bezier surfaces with an in-house Scilab script, which allows to export the dictionary used by the utility blockMesh to create the mesh for the CFD simulation. The surrogate-based optimization method here described maximizes the pump hydraulic efficiency, while keeping the total pressure rise prescribed to the design condition, in order to find the optimal impeller design. The whole optimization chain is designed for running in HPC environment with open-source software, i.e. OpenFOAM for CFD simulation, Dakota for the optimization and Scilab for the geometry parametrization.

Introduction

Centrifugal pumps can be used in different applications, requiring the operation for a wide range of pressure ratios and flow rates. The design and the performance prediction is however not an easy task due to the high number of free geometric

R. De Donno (✉) · S. Rebay · A. Ghidoni
Università degli Studi di Brescia, via Branze 38, Brescia, Italy
e-mail: r.dedonno@unibs.it

S. Rebay
e-mail: stefano.rebay@unibs.it

A. Ghidoni
e-mail: antonio.ghidoni@unibs.it

parameters to be determined, whose effect on pump performance is not trivial to determine.

Nowadays, the coupling of CFD and shape optimization algorithms represents a viable approach for an automatic, robust, and fast design of turbomachinery Van den Braembussche (2006), Pasquale et al. (2013), Guo et al. (2015), Verstraete and Alsalihi (2010), Olivero et al. (2014), Pini et al. (2014).

The aim of this study is to implement a methodology for the robust and optimal automatic design of centrifugal pumps Kim (2018), Zhou et al. (2016), Lomakin et al. (2017), Heo (2016) based on open-source software in HPC environment.

The centrifugal pump geometry chosen for this work is the well known ERCOF-TAC centrifugal pump Ubaldi et al. (1996), being available the geometry definition and the experimental results, as reported in Table 1. OpenFOAM (2018) has been used for CFD simulations, Dakota (2018) for the optimization algorithms, and an in-house Scilab (2018) script for the parametrization of the geometry which allows exporting directly the dictionary for the BlockMesh utility.

An incompressible steady-state 3D Reynolds-Averaged Navier-Stokes (RANS) approach, coupled with the RNG $k - \epsilon$ turbulence model has been used, allowing to predict reasonably well the ERCOF-TAC pump performance as demonstrated in Petit et al. (2009), Peti and Nilsson (2013). A single objective genetic algorithm (SOGA) has been applied to a surrogate model, in order to find the optimal impeller design of the ERCOF-TAC centrifugal pump for fixed operative conditions.

Table 1 Main geometric data and operating condition of the ERCOF-TAC centrifugal pump

<i>Impeller</i>	
Inlet blade diameter	$D_1 = 240 \text{ mm}$
Outlet diameter	$D_2 = 420 \text{ mm}$
Blade span	$b = 40 \text{ mm}$
Number of blades	$z_i = 7$
<i>Diffuser</i>	
Inlet vane diameter	$D_3 = 444 \text{ mm}$
Outlet vane diameter	$D_4 = 664 \text{ mm}$
Vane span	$b = 40 \text{ mm}$
Number of vanes	$z_d = 12$
<i>Operating conditions</i>	
Rotational speed	$n = 2000 \text{ rpm}$
Flow rate coefficient	$\phi = 0.048$
Total pressure rise coefficient	$\psi = 0.65$
Reynolds number	$Re = 6.5 \cdot 10^5$
<i>Inlet air reference conditions</i>	
Temperature	$T = 298 \text{ K}$
Air density	$\rho = 1.2 \text{ kg/m}^3$

Method

3D-geometry Parameterization

In order to properly define the input variables for the optimization process, the impeller of the ERCOFTAC centrifugal pump is re-expressed from data-points Ubaldi et al. (1996), Peti and Nilsson (2013) to Bezier polynomials Piegł and Tiller (1997) by means of Scilab (2018):

$$R(u) = \sum_{i=0}^n B_{n,i}(u)V_i \quad , \quad 0 \leq u \leq 1 \quad (1)$$

where $B_{n,i}(u)$ represent the Bernstein basis polynomials, u is the independent variable and V_i are the control points.

Starting from the prescribed data points $R(u)$ and choosing the Bezier polynomial order, it is possible to compute the control points of the curve that best approximate the given data-points Cho et al. (2012).

The hub and tip meridional curves are both defined by fourth order Bezier polynomials as shown in Figure 1.

The blade profile of the ERCOFTAC impeller is two-dimensional and given by data-points in the $r-\theta$ reference frame for the pressure side and the suction side Ubaldi et al. (1996). The camber line is computed from the suction side and the pressure side and then extruded along the span-direction in order to form the camber surface of the blade.

Once the camber surface is defined, it has been transformed in a parametric geometry by means of Bezier surface of third order in the two independent variables u and v . The choice of giving degrees of freedom in the span-direction even if the original impeller is purely two-dimensional, is made for enabling the optimization algorithm to twist the blade along the span-direction. The pressure side and the suction side of the blade are obtained by adding the original thickness distribution to the parametric camber surface.

The control points of hub, tip and camber surfaces of the blade are shown in Table 2 and represented in Figs. 1 and 2.

Flow Computation

Previous works in the literature Petit et al. (2009), Peti and Nilsson (2013) show that steady numerical simulation with the $k - \epsilon$ turbulence model is reasonably in good agreement with the measurements, although it obviously does not predict the unsteady features of the flow. Since an unsteady numerical simulation setup would be

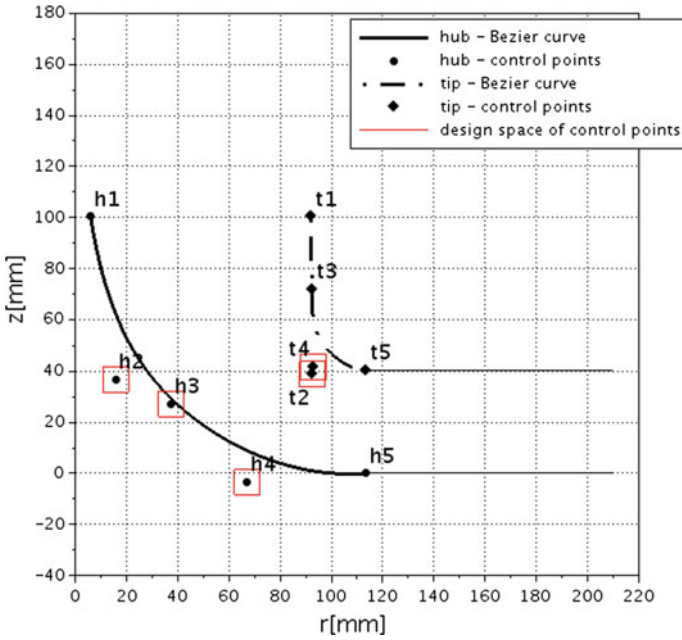


Fig. 1 Control points and design space of the impeller hub (h) and tip (t)

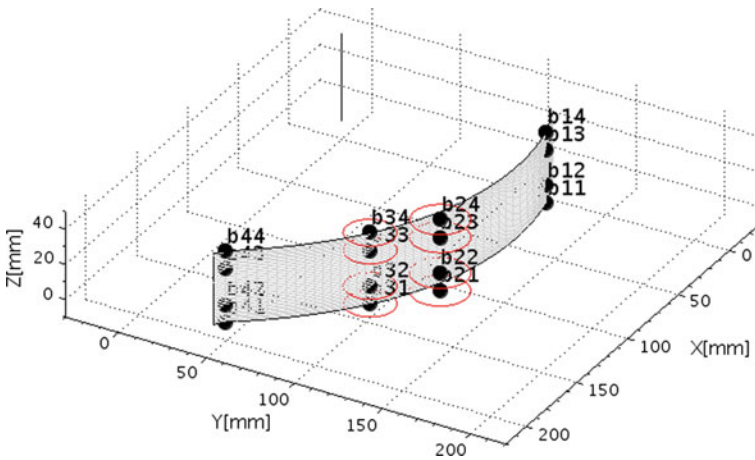


Fig. 2 Control points and design space of the impeller blade

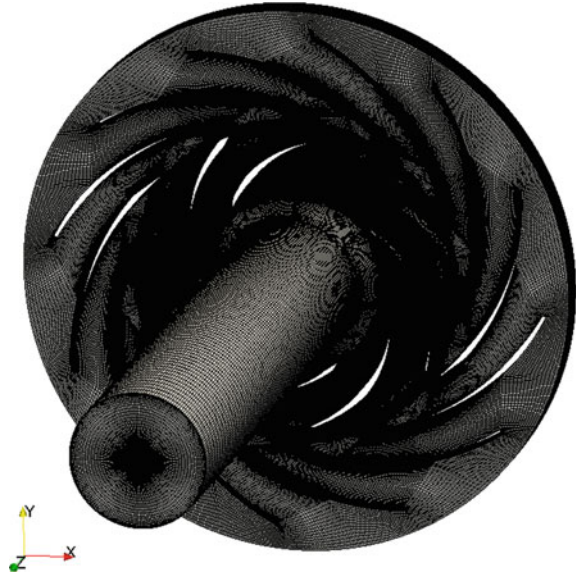
Table 2 Control points of the original geometry. The radial and axial dimensions are r and z respectively, while β indicates the angle between the radial dimension and the camber line Van den Braembussche (2006)

	r (mm)	z (mm)	β (rad)
h1	5.823	100.458	0.000
h2	15.674	36.804	0.000
h3	37.435	27.232	0.000
h4	67.098	-3.431	0.000
h5	113.630	0.000	0.000
t1	92.000	100.458	0.000
t2	92.647	38.801	0.000
t3	92.496	71.738	0.000
t4	92.814	41.709	0.000
t5	113.630	40.400	0.000
b11	128.630	0.000	0.957
b21	173.549	0.000	0.678
b31	180.023	0.000	0.541
b41	205.720	0.000	0.237
b12	128.630	10.100	0.955
b22	173.549	10.100	0.677
b32	180.023	10.100	0.540
b42	205.720	10.100	0.237
b13	128.630	30.300	0.944
b23	173.549	30.300	0.671
b33	180.023	30.300	0.535
b43	205.720	30.300	0.235
b14	128.630	40.400	0.935
b24	173.549	40.400	0.666
b34	180.023	40.400	0.530
b44	205.720	40.400	0.233

very time consuming for an optimization study and therefore would fail the objective of this work, the steady-state setup is chosen.

The incompressible steady-state 3D Reynolds-Averaged Navier-Stokes (RANS) equations, coupled with the RNG $k - \epsilon$ turbulence model are numerically solved through the open-source CFD toolbox OpenFOAM (2018) with the high accuracy numerical schemes available in the literature Auvinen et al. (2010). The multiple frame of reference approach is adopted together with the frozen rotor technique and the fluxes at the interface are transferred using the General Grid Interface (GGI) Beaudoin and Jasak (2008), Jasak (2011), Petit et al. (2009).

Fig. 3 Grid for the CFD calculation



The computational grid includes the inlet, the rotor and the vaned diffuser regions meshed separately with the OpenFOAM utility blockMesh. The grids of the inlet and of the vaned diffuser do not change during the optimization process and therefore are built only once. The complete grid is composed of about 4 million hexahedral cells and the fluid domain of the initial design is shown in Figure 3, while Figure 4 shows a detail of the grid in the impeller blade region.

To match the available experimental measurement conditions Ubaldi et al. (1996), the impeller rotational speed is set to 2000 rpm and an inlet axial velocity corresponding to a flow coefficient ϕ equal to 0.048 is prescribed, where

$$\phi = 4Q / (U_2 \pi D_2^2), \quad (2)$$

Q is the flow rate, U_2 is the peripheral velocity at the impeller outlet and D_2 is the diameter at the impeller outlet. The inlet and outlet boundary conditions are set according to the simulations already performed on this case-study and available in the literature Peti and Nilsson (2013).

The computational quantities of the CFD simulations are the efficiency η and the total pressure rise coefficient ψ , where

$$\psi = 2(p_{t4} - p_{t0}) / \rho U_2^2, \quad (3)$$

p_{t4} is total pressure at the diffuser outlet and p_{t0} is total pressure at the suction pipe. The mass averaged values of η and ψ calculated between inflow and outflow for the initial design are 0.63 and 0.60, respectively.

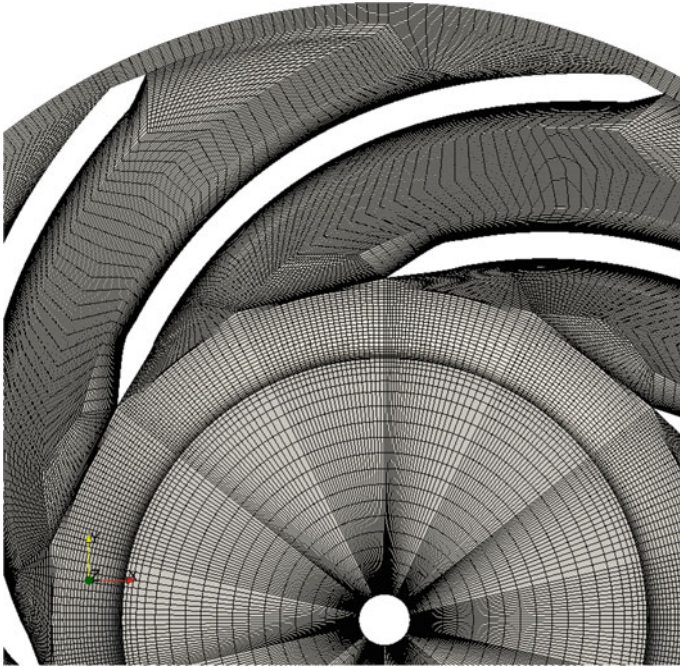


Fig. 4 Grid for the CFD calculation, detail of the impeller blade

Figures 5 and 6 show the comparison between the experimental measurements Ubaldi et al. (1996) and the CFD results with grid and setup described above, in terms of radial and tangential velocity distribution near the impeller outflow. The average relative error in the x and y axis is lower than 30%, showing a reasonable agreement between the steady-state numerical setup and the unsteady nature of the flow. In fact the experimental trend of the velocity distributions is well captured by the simulations.

Optimization Strategy

In the centrifugal pump performance, the hydraulic efficiency η and the total pressure rise coefficient ψ have a fundamental role and therefore are chosen as optimization objective and constraint, respectively. In particular the optimization algorithm maximizes η , while keeping ψ constrained to the operative point analyzed with a tolerance of $\pm 5\%$.

In this work the single objective genetic algorithm SOGA, available in the software Dakota (2018), is applied to a surrogate model, in order to find the global optimum of the objective function Coello et al. (2007), Pierret and Van den Braembussche (1999).

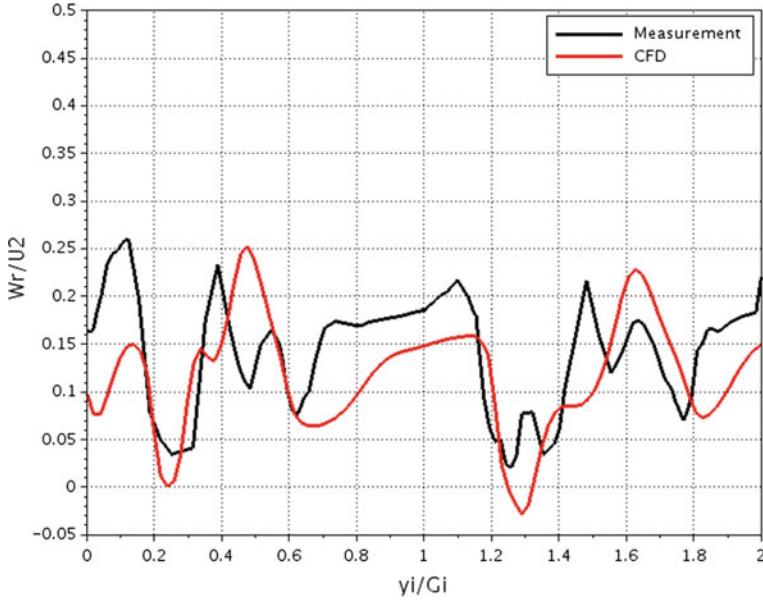


Fig. 5 Instantaneous distributions of the ensemble averaged radial velocity at the impeller outlet, at midspan position. Relative position between impeller and diffuser blade at $t/T_i=0.146$ according to Ubaldi et al. (1996)

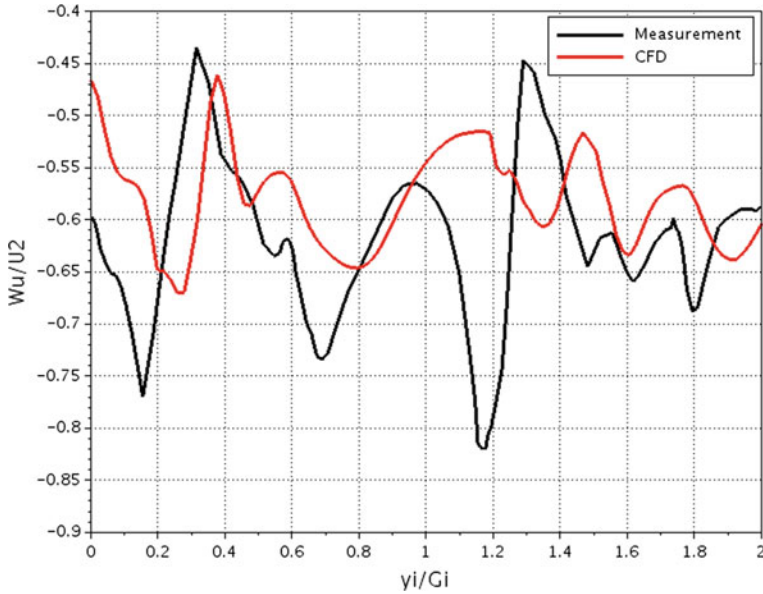


Fig. 6 Instantaneous distributions of the ensemble averaged tangential relative velocity at the impeller outlet, at midspan position. Relative position between impeller and diffuser blade at $t/T_i=0.126$ according to Ubaldi et al. (1996)

The surrogate model has been built on the results of 170 computer experiments (ten times the input variables number), generated through the Latin Hypercube Sampling (LHS) method.

The surrogate-based optimization strategy of this work consists in the following steps:

- optimization problem definition (design variables, objectives and constrained),
- computational design of experiments by means of the LHS method,
- surrogate model construction,
- surrogate model accuracy evaluation through cross validation,
- calculation of further designs if the surrogate accuracy is not satisfying,
- constrained single objective genetic algorithm on the accurate surrogate model,
- CFD simulation of the optimum design.

Two surrogates, Kriging and Artificial Neural Network, have been compared regarding their accuracy respect to the CFD simulations. The analysis has been carried out on the 5% of the computer experiments, showing that the two meta-models fit the efficiency similarly, while the total pressure rise coefficient is fitted one order magnitude better by the Kriging meta-model. Therefore the Kriging surrogate model, showing a better accuracy for the problem analyzed, has been used for this study.

The population size of the genetic algorithm has been set equal to 100 samples, while the mutation rate and the crossover rate have been set equal to 0.1 and 0.8, respectively.

Figures 1 and 2 show the design space of the input variables for hub, tip and blade. In order to prevent changes in the overall size of the centrifugal pump, the initial and final control points of hub and tip as well as the control points of leading edge and trailing edge of the blades are not modified during the optimization process. Furthermore control point t3 shown in Fig. 1 is not taken into account for the optimization for the sake of shape feasibility. The input variable b1-beta controls the β -coordinate of control points b21, b22, b23 and b24 while b2-beta controls the β -coordinate of control points b31, b32, b33 and b34. The input variables b1-r, b2-r, b3-r and b4-r control the r-coordinates of control points b21-b31, b22-b32, b23-b33 and b24-b34 respectively. Beyond Bezier control points, also the impeller blade number is considered as input variable. A total number of 17 design variables is therefore considered in this study, details are shown in Table 3.

Results and Discussions

The relation between the objective function η and the nonlinear inequality constrained ψ is shown in Figure 7, where the output of the computer experiments and the evolution of the genetic algorithm applied to the surrogate model are represented. The figure highlights also the improvement from the initial to the optimal design in terms of pump efficiency.

Table 3 Input variables description: original value, space domain and values of the optimal design. The letter h indicates the hub, t indicates the tip and b indicates the blade. R, z and β indicate the coordinate in the reference frame

	Descr.	Orig.	Min	Max	Type	Unit	Opt.
1	h2-r	0.000	-5.000	5.000	float.	mm	-4.971
2	h3-r	0.000	-5.000	5.000	float.	mm	1.260
3	h4-r	0.000	-5.000	5.000	float.	mm	-4.846
4	h2-z	0.000	-5.000	5.000	float.	mm	-4.961
5	h3-z	0.000	-5.000	5.000	float.	mm	4.976
6	h4-z	0.000	-5.000	5.000	float.	mm	1.251
7	t2-r	0.000	-5.000	5.000	float.	mm	4.930
8	t4-r	0.000	-5.000	5.000	float.	mm	4.902
9	t2-z	0.000	-5.000	5.000	float.	mm	4.917
10	t4-z	0.000	-5.000	5.000	float.	mm	-1.647
11	b1-beta	0.000	-0.050	0.050	float.	rad	-0.035
12	b2-beta	0.000	-0.050	0.050	float.	rad	-0.023
13	b1-r	0.000	-5.000	5.000	float.	mm	-4.923
14	b2-r	0.000	-5.000	5.000	float.	mm	4.978
15	b3-r	0.000	-5.000	5.000	float.	mm	-2.070
16	b4-r	0.000	-5.000	5.000	float.	mm	4.858
17	n-blades	7	5	9	int.	-	6

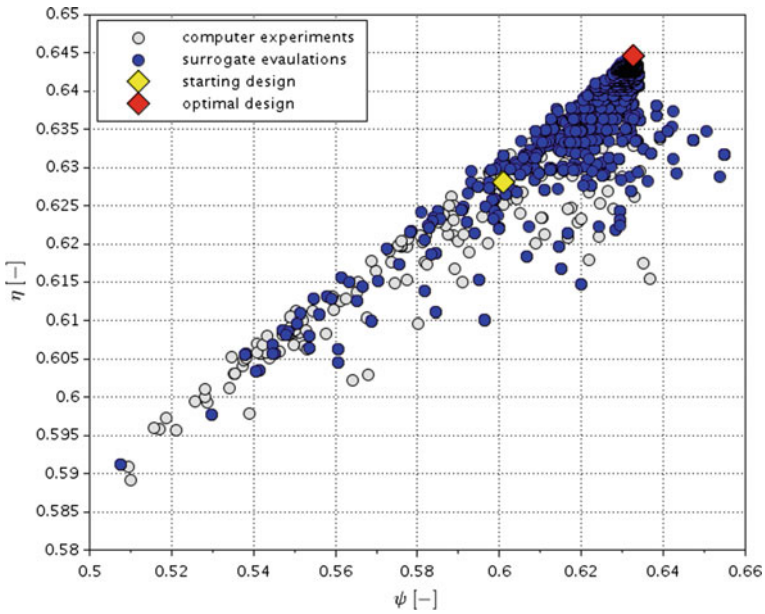


Fig. 7 Optimization results: SOGA applied to the Kriging surrogate model

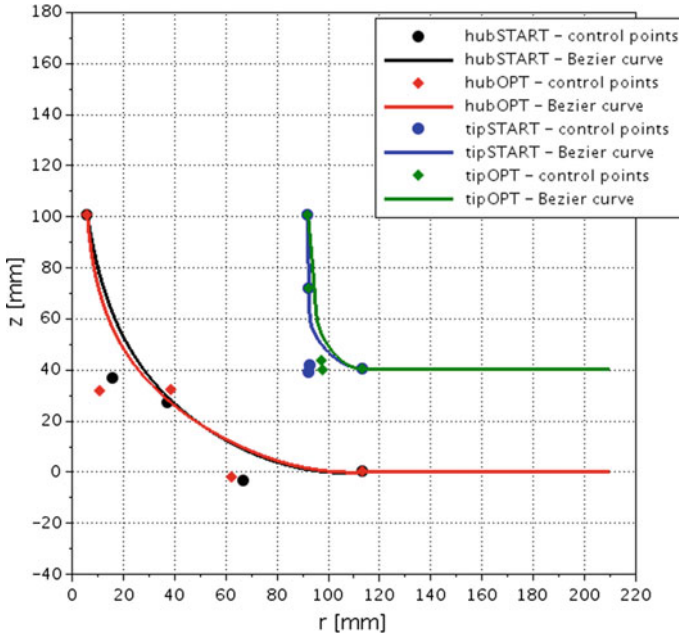


Fig. 8 Comparison between the starting and the optimal design. Detail of the impeller hub and tip

Figures 8 and 9 show the geometrical comparison between the starting and the optimal design for the hub, tip and blade. It can be observed that the hub and tip have been modified in order to increase the flow passage before the blade leading edge, while the blade surface has been twisted along the span-direction. After the optimization, the impeller blade curvature has been reduced and gradually increases when moving from the impeller hub to the impeller tip as shown in Fig. 10. It is also interesting to notice that the optimal design reduces the number of blades to 6 instead of 7. Table 3 shows the design variables comparison between the starting and the optimal design.

To better understand the effects of the geometry changes on the pump performance, the static pressure coefficient C_p as well as the wall shear stress τ_w have been calculated for the impeller blade and for the diffuser blade at different span positions. The static pressure coefficient is defined as

$$C_p = 2(p - p_0) / \rho U_2^2, \tag{4}$$

where p is the static pressure of a generic point and p_0 is the static pressure in the suction pipe, while the wall shear stress is calculated as

$$\tau_w = \mu \left(\frac{\partial u}{\partial y} \right)_{y=0}, \tag{5}$$

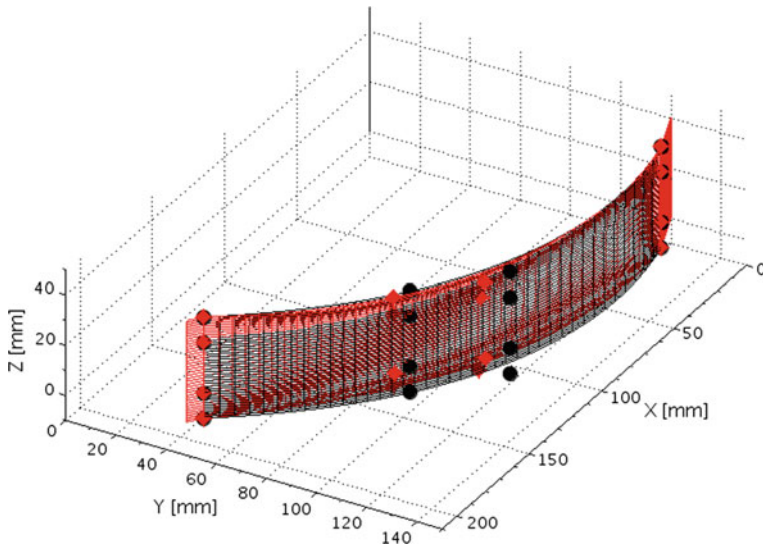


Fig. 9 Comparison between the starting and the optimal design for the impeller blade. The black color represents the starting design, while the red color represents the optimal design. The markers show the Bezier surface control points

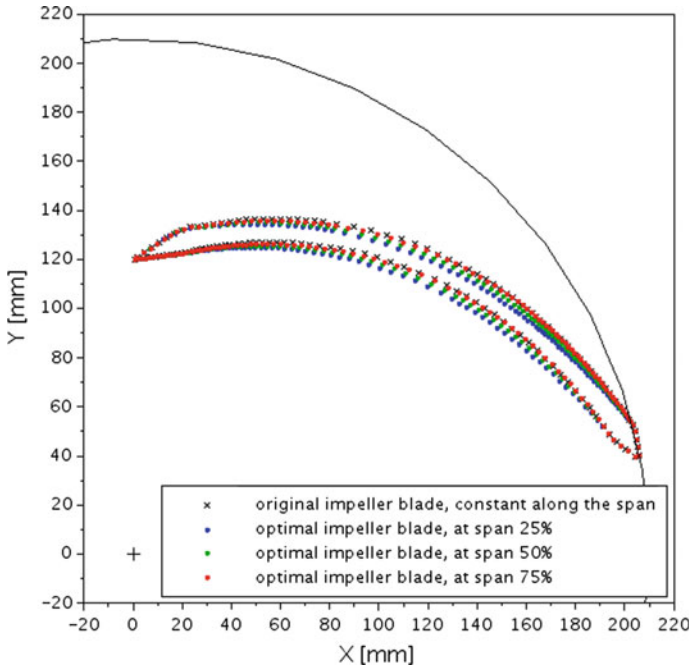


Fig. 10 Impeller shape comparison between the original and the optimal design at different sections along the span direction

where μ is the dynamic viscosity, u is the flow velocity parallel to the wall and y is the distance to the wall.

The impeller geometry, even with different number of blades, is built keeping fixed the relative position between one impeller blade and the diffuser blades, with respect to the initial configuration. This choice allows a direct comparison of the results for the initial and optimal geometry, Figure 11 shows the blades of impeller and diffuser where the results are calculated.

The comparison of the static pressure coefficient shows that the optimal impeller has an higher fluid-dynamic load at all the span positions, as shown in Figs. 12, 14 and 16, without modifying the overall behavior of the wall shear stress depicted in Figs. 13, 15 and 17. These improvements and the decrease of the blades number, i.e. a reduction of the wetted area, explain the better efficiency of the optimal design.

The higher fluid-dynamic load of the impeller blades in the optimal design can be observed also in Fig. 19, where the high-velocity region at the impeller pressure and suction side is significantly greater with respect to the original design, shown in Fig. 18. The increase of velocity at impeller outlet and, consequently, at the diffuser inlet explains the higher peak of the wall shear stress of the optimal design diffuser blade shown in Fig. 20. However Fig. 20 shows that the τ_w distribution on the diffuser blade slightly changes, suggesting that the overall performance of the diffuser has not been modified during the optimization process.

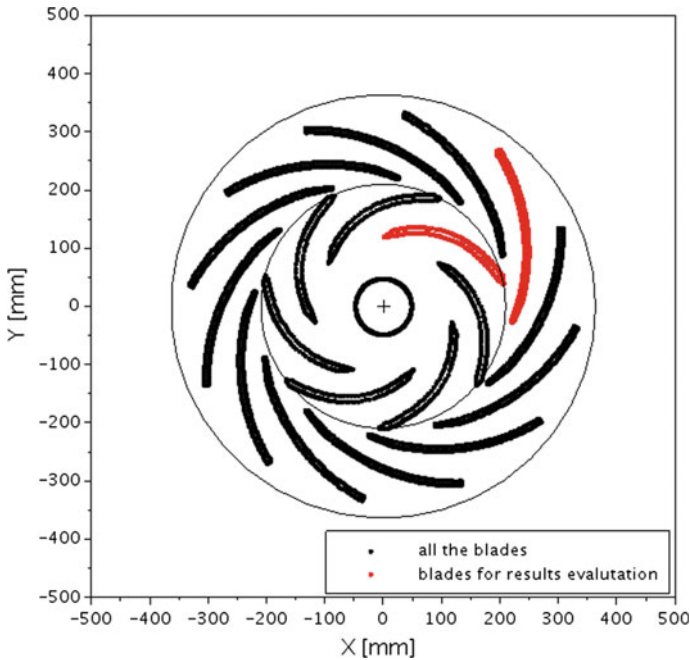


Fig. 11 Blades of impeller and diffuser where the static pressure coefficient and the wall shear stress are calculated

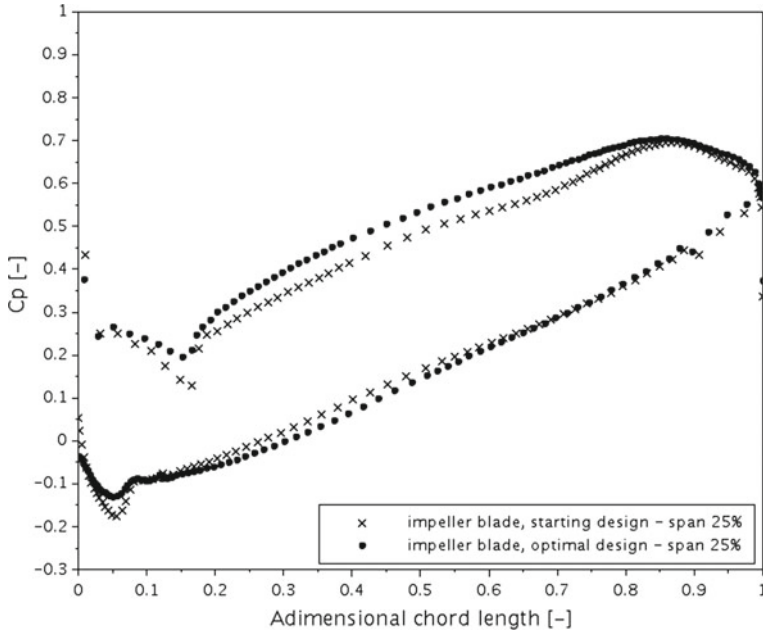


Fig. 12 C_p distribution on the impeller blade at span 25% for the original and optimized configuration

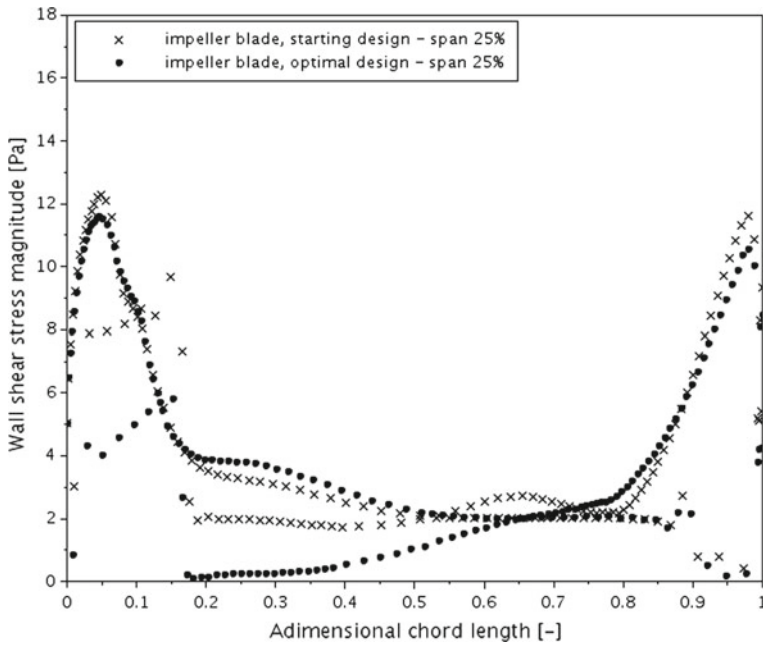


Fig. 13 τ_w distribution on the impeller blade at span 25% for the original and optimized configuration

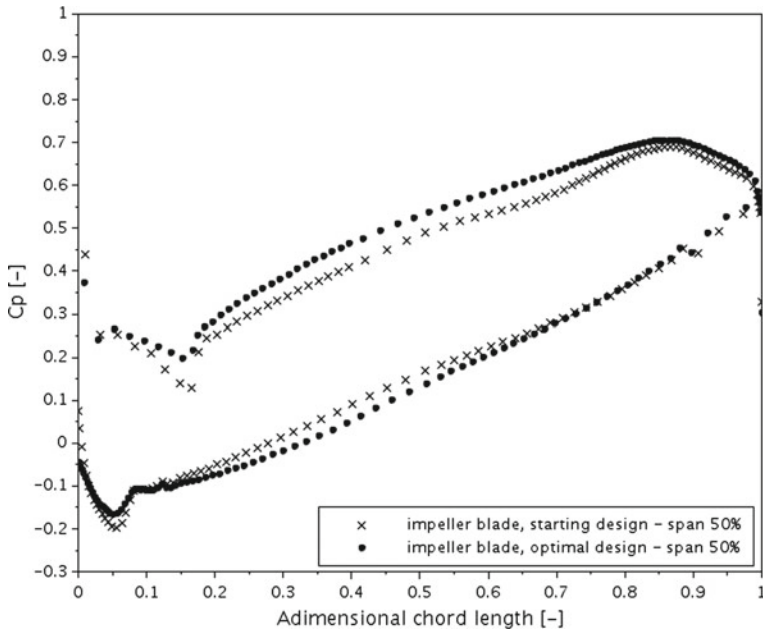


Fig. 14 C_p distribution on the impeller blade at span 50% for the original and optimized configuration

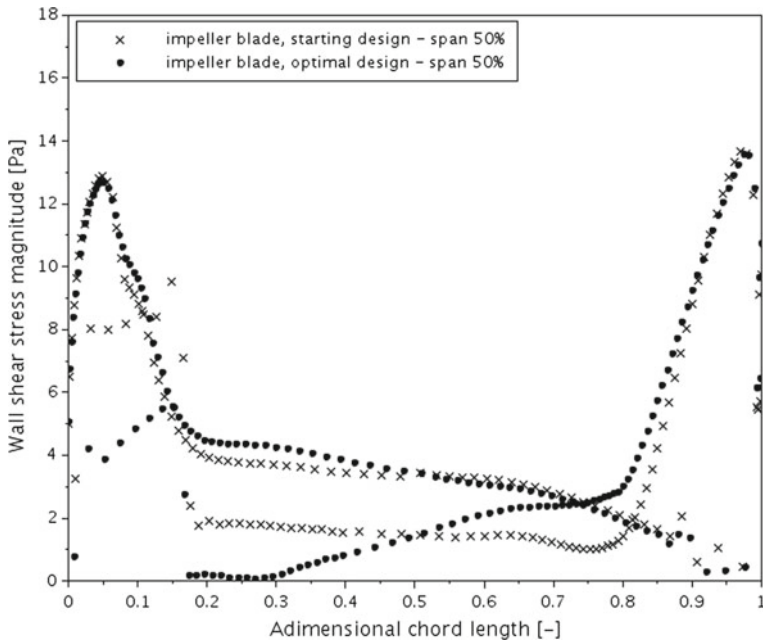


Fig. 15 τ_w distribution on the impeller blade at span 50% for the original and optimized configuration

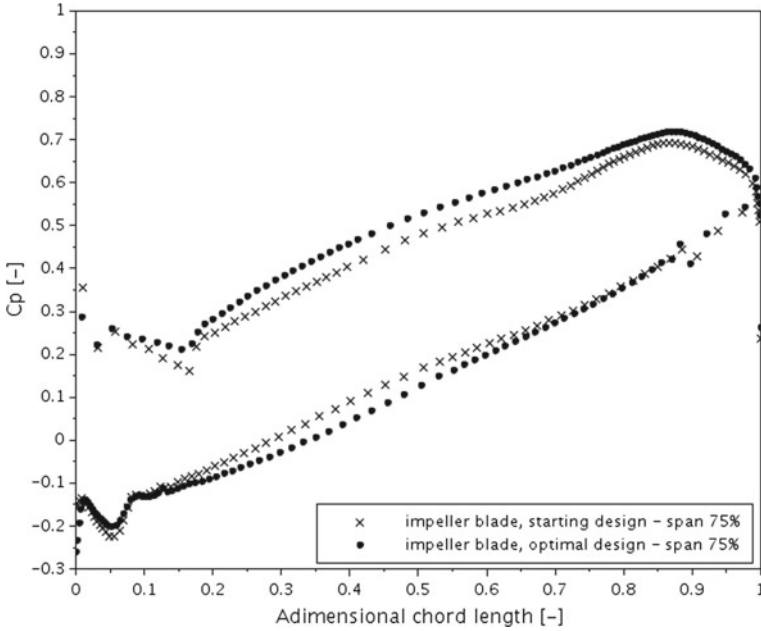


Fig. 16 C_p distribution on the impeller blade at span 75% for the original and optimized configuration

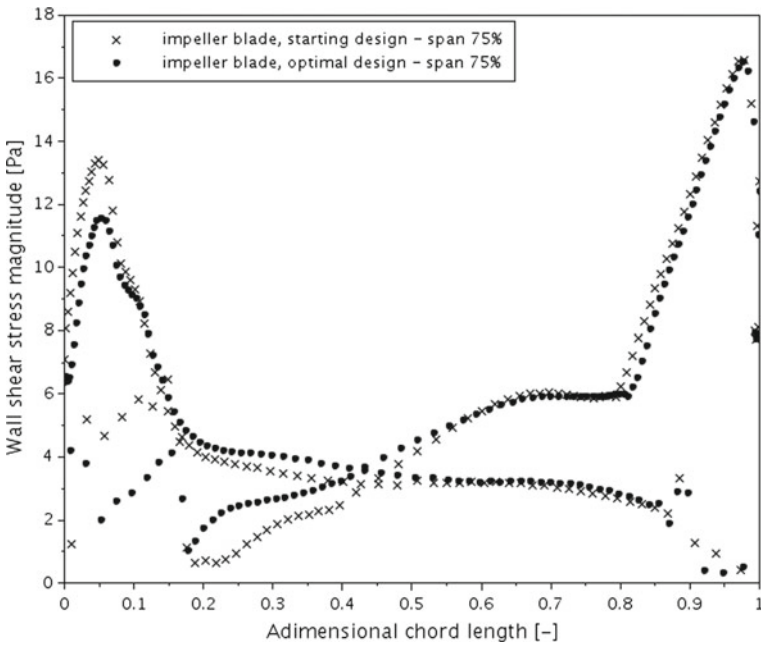


Fig. 17 τ_w distribution on the impeller blade at span 75% for the original and optimized configuration



Fig. 18 Velocity magnitude contours of the original impeller at span 50%

Conclusions

A fully automated surrogate-based optimization method has been presented for improving the centrifugal pump impeller efficiency, entirely based on open-source and in-house software.

The method has been tested on the ERCOFTAC centrifugal pump, where the impeller shape has been converted in Bezier polynomials from data points and 17 control points have been used as design variables for the optimization.

The Kriging surrogate model has been adopted for this work and trained on computer experiments in order to connect accurately the impeller geometry with the pump performance, predicted by computational fluid dynamics. A single objective genetic algorithm has been set in order to maximize the pump efficiency coefficient η , while keeping constrained the pressure rise coefficient ψ , for making the pump working at the initial operative point.



Fig. 19 Velocity magnitude contours of the optimal impeller at span 50%

The results of this work show an improvement of the pump efficiency about 2.63% with respect to the initial design and, therefore, demonstrate the effectiveness of a surrogate-based optimization strategy for improving the pump hydraulic efficiency, while maintaining the prescribed operative condition.

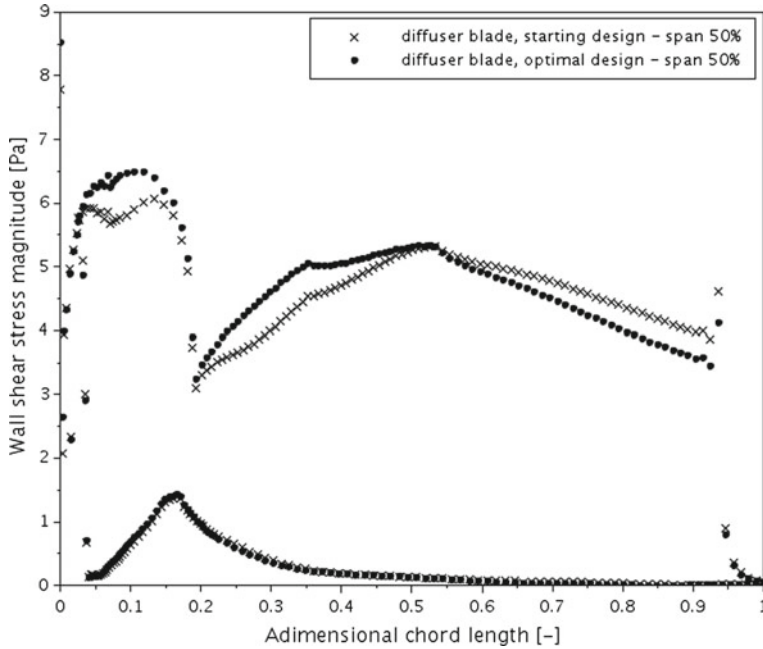


Fig. 20 τ_w distribution on the diffuser blade at span 50% for the original and optimized configuration

References

- Auvinen M, Ala-Juusela J, Nicholas P, Siikonen T (2010) Time-accurate turbomachinery simulations with open-source CFD; Flow analysis of a single-channel pump with OpenFOAM. In: Pereira JCF, Sequiera A, Pereira JMC, (eds) Fifth european conference on computational fluid dynamics, Ecomas CFD 2010, Lisbon, Portugal, 14.-17.6.2010. Lisbon, Portugal
- Beaudoin M, Jasak H Development of a generalized grid interface for turbomachinery simulations with OpenFOAM. Proceedings of the Open Source CFD International Conference, Berlin, Germany (2008)
- Cho S-Y, Ahn K-J, Lee Y-D, Kim Y-C (2012) Optimal design of a centrifugal compressor impeller using evolutionary algorithms. *Math Problems Eng*
- Coello CA, Lamont GB, Veldhuizen D (2007) *Evolutionary algorithms for solving multi-objective problems*. Springer
- Dakota 6.3: <https://dakota.sandia.gov/>
- Guo Z, Song L, Zhou Z, Li Y, Feng Z (2015) Multi-objective aerodynamic optimization design and data mining of a high pressure ratio centrifugal impeller. *ASME J Eng Gas Turbines Power* 137(9):092602–092602–14
- Heo M, Ma S, Shim H, Kim K (2016) High-efficiency design optimization of a centrifugal pump. *J Mechanical Sci Technol* 30(9):3917–3927
- Jasak H, Beaudoin M, OpenFOAM turbo tools: from general purpose CFD to turbomachinery simulations, fluids engineering division summer meeting, ASME-JSME-KSME Joint Fluids Engineering Conference (2011)

- Kim J, Kim K (2012) Analysis and optimization of a vaned diffuser in a mixed flow pump to improve hydrodynamic performance. *ASME J Fluids Eng* 134(7):071104–071104-10
- Lomakin VO, Chaburko PS, Kuleshova MS (2017) Multi-criteria optimization of the flow of a centrifugal pump on energy and vibroacoustic characteristics. *Procedia Eng* 176:476–482
- Olivero M, Pasquale D, Ghidoni A, Rebay S (2014) Three-dimensional turbulent optimization of vaned diffusers for centrifugal compressors based on metamodel-assisted genetic algorithms. *Optim Eng* 15(4):973–992
- OpenFOAM Extend 3.2: <https://sourceforge.net/projects/openfoam-extend/>
- Pasquale D, Ghidoni A, Rebay S (2013) Shape Optimization of an organic rankine cycle radial turbine nozzle. *ASME J Eng Gas Turbines Power*, 135(4):042308–042308-13
- Peti O, Nilsson H (2013) Numerical investigations of unsteady flow in a centrifugal pump with a vaned diffuser. *Int J Rotating Machinery*
- Petit O, Page M, Beaudoin M, Nilsson H (2009) The ERCOFTAC centrifugal pump OpenFOAM case-study. In: 3rd IAHR international meeting of the workgroup on cavitation and dynamic problems in hydraulic machinery and systems
- Piegl LA, Tiller W (1997) *The NURBS book*. Springer
- Pierret S, Van den Braembussche RA (1999) Turbomachinery blade design using a navier-stokes solver and artificial neural network. *ASME. J. Turbomach.* 121(2):326–332
- Pini M, Persico G, Pasquale D, Rebay S (2014) Adjoint Method for shape optimization in real-gas flow applications. *J Eng Gas Turbines Power* 3(137):032604–032604-13
- Scilab 5.5.2: <http://www.scilab.org>
- Ubaldi M, Zunino P, Barigozzi G, Cattanei A (1996) An experimental investigation of stator induced unsteadiness on centrifugal impeller outflow. *ASME J Turbomach* 118(1):41–51
- Van den Braembussche RA (2006) Optimization of radial impeller geometry. RTO-EN-AVT-143
- Verstraete T, Alsalihi Z, Van den Braembussche RA (2010) Multidisciplinary optimization of a radial compressor for microgas turbine applications. *ASME J Turbomach* 132(3):031004–031004-7
- Zhou L, Shi W, Wu S (2016) High-efficiency design optimization of a centrifugal pump. *J Mechanical Sci Technol* 30(9):3917–3927

CFD Based Design Optimization of a Cabinet Nitrogen Generator



Bárbara Arizmendi Gutiérrez, Edmondo Minisci and Greig Chisholm

Abstract The design of mechanical enclosures is evolving to be more compact and quieter and this compromises the cooling of the internal components. Computational Fluid Dynamics (CFD) based optimization could significantly improve the cooling efficiency of the critical parts of the components to ensure their performance and reliability. This work presents the CFD surrogate based optimization of the forced cooling of two reciprocating compressors located in an enclosure from a gas generator. Due to the challenging project time constraints, the accuracy of the results was compromised to make optimization feasible. The parameters to be optimized were related to the position of the compressors and the cooling fans. The boundary conditions associated to the cooling of the critical parts were derived by experimental data. Artificial Neural Networks (ANNs) were used to construct a surrogate model of the computational model to reduce the time and resources required. The combination of the ANN model with a multi start-gradient based algorithm optimized the position of compressors and cooling fans to minimize the average temperature on the critical parts. A set of new enclosure designs were found with outstanding CFD based performance compared with the design elaborated by engineering intuition.

B. A. Gutiérrez (✉) · E. Minisci
University of Strathclyde, 16 Richmond Street, G1 1XQ Glasgow, Scotland
e-mail: barbaraarizmendi@strath.ac.uk

E. Minisci
e-mail: edmondo.minisci@strath.ac.uk

G. Chisholm
Peak Scientific Instruments Ltd, 11 Fountain Crescent, Paisley, PA4 9RE Renfrew, Scotland
e-mail: gchisholm@peakscientific.com

Introduction

The design of mechanical enclosures is evolving to be more compact and quieter thus decreasing in size and aperture areas. This compromises their cooling performance, potentially causing higher temperatures in the compartment ambient air and the internal components, and increasing hotspot temperatures in temperature sensitive components and parts. Surrogate based CFD optimization could play a key role in delivering designs that minimized the temperature of those critical parts. For this industrial project, reliability improvements of 2 two-stages reciprocating compressors from a gas generator were sought by reducing temperatures in their identified critical parts which were their sleeves. The sleeves are the cylinders where the compression process occurs and are located on the top of the compressors.

The thermal interaction between heat sources located within enclosures as well as the selection of inappropriate cooling means causes higher temperatures in the heat sources which might compromise the reliability of temperature sensitive components. It is essential to ensure the cooling means can remove all the heat generated by the sources and the adequate positioning of heat sources and cooling means (i.e. fans) if those are needed. In this case, airflow delivered by the fans was found to be able to remove the heat generated by the compressors in operation. There are research efforts (Chen and Liu 2002; Soleimani et al. 2011; Madadi and Balaji 2008; Sudhakar et al. 2010; Kadiyala and Chattopadhyay 2011; Hotta and Venkateshan 2015; Dias and Milanez 2006; Sudhakar et al. 2010) on the positioning of several heat sources within an enclosure applied to electronics cooling by means of free and forced convection and to the design of heat exchangers. Electronic components are decreasing in size and consequently increasing in power density leading to higher peak temperatures. The research in this area focuses on decreasing the peak temperatures on the heat sources by changing their position or the heat distribution for each heat source. Furthermore, for heat exchangers design, the positioning of thermal insulation in tubes was investigated to obtain an even temperature profile across the inner surface of the tubes for homogeneous heat transfer to the fluid (Sudhakar et al. 2009). Several authors have sought the optimality, that is finding the position of the sources of heat or the heat distribution in the sources that minimises the surface peak temperature. The design parameters investigated in both cases are the coordinates that determine the position of the sources within the enclosure or the heat load of each source. The design performance has been assessed experimentally, analytically and computationally by means of CFD modelling.

Chen and Liu experimentally proved intuitive equidistant positioning placed horizontally within a compartment and cooled by forced convection was not the optimal in terms of cooling performance (Chen and Liu 2002). This justifies the investigation of the positioning of the components in order to find the distribution that maximises the cooling efficiency of the heat sources.

Since the cooling of the heat sources is a fluid dynamic process, it can be modelled by means of CFD codes. The location of the heat sources can be parameterized and optimised to minimise the peak temperatures. For instance, Soleimani optimised the

position of two discrete heat sources placed within a vertical enclosure and cooled by free convection (Soleimani et al. 2011). The optimization algorithm selected was the Particle Swarm and it required 200 CFD runs to accomplish the global minimum. This methodology could be applied because there were 2 design variables and the model was very simple.

For more complex problems whose resolution and number of design variables is higher, the construction of a surrogate model is mandatory be able to explore more thoroughly the design space and find the best performing design possible. This approach reduces the number of CFD calculations required. The preferred method for the construction of surrogate models for this application was found to be Artificial Neural Networks (ANN's) and the surrogate model constructed was optimised by means of Genetic Algorithms (GA) (Madadi and Balaji 2008; Sudhakar et al. 2010; Kadiyala and Chattopadhyay 2011; Hotta and Venkateshan 2015). For instance Madadi optimised the position of 3 discrete heat sources within a 2D vertical enclosure cooled by forced convection (Madadi and Balaji 2008). The identified design parameters were related to the position of the sources. The authors integrated an ANN with a GA to find the design that minimised the peak temperature on the surface of the sources. The implementation of the methodology found a design that computationally reduced the peak temperature from an initial design by 22.71 °C. Although the results seemed very promising, they were not validated experimentally.

Sudhakar optimised the heat distribution of 9 heat sources within a horizontal cavity and cooled by means of forced convection (Sudhakar et al. 2010). In this case, an ANN was coupled with a GA again to find the optimal distribution of heat in the sources. The computational optimum obtained minimised the peak temperature measured on the geometrical centre of the sources surface. These results were validated experimentally accomplishing a temperature decrease of 10.30 °C with respect to the uniform heat distribution. All the temperatures measured over each one of the heat sources were decreased, finding a more favourable thermal profile. These results gave validity to the implementation of this technique in the accomplishment of more effective positioning of individual and constant heat sources located within an enclosure.

In the literature reviewed, the heat sources were treated as individual, independent and constant. For this application, the sleeves of the compressor were treated similarly and their position within the enclosure was investigated to minimise their surface temperatures. The sleeves are cooled by 2 inlet and 2 outlet axial fans which provide a pressure rise and swirling to the airflow. The optimum position of the fans was also investigated.

In the first part, the background of the project and a description of the design of the cooling system are presented. Next, the methodology is introduced including the construction of the CFD model from experimental test and the CFD surrogate based optimization approach. Next, it is presented a description of the design parameters and cost function selected as well as the construction of the surrogate and the optimization algorithm implemented. Then, a description of the experimental methodology for the validation of the results is described. Finally, the results obtained from the CFD

based optimization and the experimental validation are reported and discussed and the conclusions from this study and the future work are drawn.

Background of the Project

The mechanical enclosure where the 2 two-stages reciprocating compressors are located is a part of a gas generator that consists of a large cabinet with two side doors where all the components for the gas generation process are located. The aforementioned mechanical enclosure is placed in the top back side of the generator. Those compressors are cooled by two inlet axial fans which draw air from inside the gas generator and two outlet fans that push the hot air outside the generator. Within the compressor compartment, there are other required components as well as wiring and tubing. Each compressor has 2 sleeves, high pressure and low pressure ones which are cylindrical element where the compression of the ambient air takes place.

This project has been carried out as a case study to introduce an industrial partner to CFD based optimization within the framework of the activities of a Knowledge Transfer Partnership (KTP) between the Mechanical and Aerospace Engineering department of the University of Strathclyde and Peak Scientific Instruments Ltd. As such, computational resources and available time were significantly limited therefore, simplifications were mandatory.

Methodology

The approach started from the implementation of a simplified CFD modelling of the enclosure, still able to predict cooling improvement trends on the sleeves requiring a minimum amount of computational resources and wall time to meet the project timings. First, experimental tests were conducted to understand the thermal behavior of the compressors as it was initially unknown. Next, a CFD model was constructed based on those tests for a qualitative representation of the cooling of the sleeves and a surrogate model was created to predict the cooling trends. After, the position of the compressors and fans was optimized to find a design with more efficient cooling.

Initial Experimental Tests

The crucial features of the model were the airflow delivered by the inlet cooling fans and the thermal behavior of the sleeves. The experimental tests aimed to get a general understanding of the system, but particularly they were focused on getting the maximum information possible from those key features. To locate the hottest parts of the compressors, a thermal image of the compressors was taken with a

thermal imaging camera (FLIR SC500). The temperatures were calibrated with thermocouple measurements in 2 locations over the surface of the compressors.

Next, the temperatures at 8 discrete locations of the critical parts were measured with type K thermocouples. The ambient and the inlet and outlet temperatures were monitored. These tests provided a quantitative insight into the range of temperatures found on the sleeves and in the compartment.

The velocity of the airflow delivered by the fans was measured at 8 discrete locations of the inlet fans with a hot wire anemometer. For each location, 5 different directions were sampled. The outcome was a quantification of the air velocity magnitudes in the area right next to the inlet fans.

CFD Modelling

The geometry considered was based on the dedicated compressors compartment, neglecting any interaction with the rest of the gas generator. Geometric details of the compressors such as nuts, bolts and clefts were not modelled to reduce the number of mesh cells required. The compressors heads, where the sleeves were located, were modelled with more detail to increase the resolution on the critical areas. The fan frames were modelled as well as the outlet fan grilles but the blades of the fans were not included in the model although they are represented in Figs. 1 and 4 for better visualization. All the remaining components were not represented, aiming to reduce the wall time of the calculations. The CAD geometry considered is depicted in Fig. 1. The inlet fans are found on the right side of the image and the outlet fans are positioned on the left.

To get the fluid properties in any location, the Navier-Stokes three-dimensional conservation equations for mass, momentum and energy were numerically solved through the software package (SolidWorks Flow Simulation 2016). The turbulence

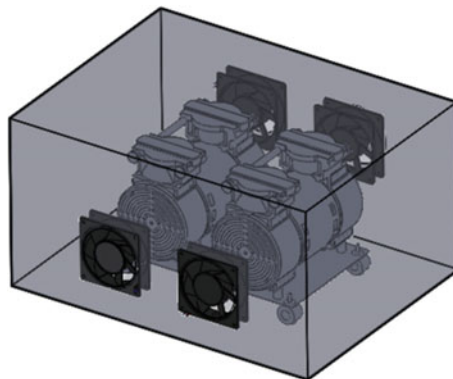


Fig. 1 Simplified CAD model of the compartment

was modelled in every energy scale by means of a Favre-Average of the equations for compressible flows and the turbulence model selected was the $k-\epsilon$ ¹.

The heat sources were modelled by constant heat flux boundary conditions on the surfaces of the compressors. A velocity boundary condition dependent on the pressure was implemented on the inlet and outlet fans with additional swirling. The fan curve and spinning velocity were obtained from the specifications of the manufacturer¹. The computational velocities delivered by the inlet cooling fans were kept within the range of the experimental measurements. The temperature of the inlet cooling air was assumed to be constant and it was taken from the experimental tests. The temperature gradient across the compartment walls was set to 0, drawing a worst-case scenario.

The mesh entailed the spatial discretization of the computational domain into cube shaped cells. The refinement consisted on quartering the mesh cell size close to solid boundaries where large temperature or velocity gradients were expected. For this project, the selection of the cell size of the mesh and consequently the accuracy of the results, was compromised by the computational budget and the run time available for optimization. The maximum computational time of the CFD model was dictated by the allocated time and the expected required number of CFD runs for optimization. That prohibited the selection of a fine mesh thus compromised the accuracy of the results. But that compromise was necessary to enable the performance of the optimization within the project timings. A mesh sensitivity study was conducted to obtain the range of magnitudes of the error induced. Five different cell sizes were tested with a size ratio of 1.2. The property tracked was the average temperature on the sleeves. The calculations modelled 7 physical seconds of a transient process. The average temperature on the sleeves was averaged over the last 2 physical seconds to eliminate turbulent fluctuations when comparing different designs. The results obtained are shown in the Table 1.

The mesh 3 was selected because it met the limits on the available runtime per simulation which was 7.31 h. By selecting this mesh, the error induced was at least of 4.29 K and this will be considered when analysing the improvements achieved by optimization. This step was crucial to enable the optimization of the cooling of the sleeves, even with the implementation of a surrogate approach, within the timings of

Table 1 Mesh sensitivity study

Mesh	Nb of cells	Computational time (s)	Temperature (K)
1	722,364	10,604	328.72
2	1,032,537	16,508	331.74
3	1,565,154	26,325	339.78
4	2,340,485	45,811	341.89
5	3,467,445	76,079	344.07

¹Product Data Sheet 9956—*EBM-Papst*.

the project. This model was used as the baseline design to assess the computational improvements obtained by the design optimization study.

CFD Surrogate Based Optimization

The cost function selected was the average temperature on all the sleeves of the compressors but the peak temperatures were tracked as well to have an additional criterion to select the best design. The design variables were relative to the position of the compressors and fans within the compartment. The fan variables were related to their distance to the sidewalls (w_1 , w_2) and bottom wall (h_1 , h_2). The compressor variables were associated to their distance to the front (l) and side (m) walls and their angle (a) with respect to the front wall. The new design was constrained to be symmetric. There was a total of 7 design variables to be investigated to minimize the average temperature on the critical components of the compressors. The bounds of the parameters, which are depicted in the Table 2, were defined in such a way that any combination of parameters produced a valid design.

The cost per evaluation of the CFD model and the number of design variables prohibited its direct optimization. Therefore, a cheaper to evaluate surrogate model was used which significantly decreased the number of CFD runs to obtain the optimal solution. ANNs have been successfully used in research to construct surrogate models for optimization (Madadi and Balaji 2008; Sudhakar et al. 2010; Kadiyala and Chattopadhyay 2011; Hotta and Venkateshan 2015). The architecture of the selected feedforward ANN consists of an input layer, a hidden layer constituted of 20 neurons and an output layer with 1 neuron. The activation function of the hidden layer neurons is a logarithmic sigmoid one and for the output layer is a linear function.

The initial weights of the nodes connections were randomly selected. The algorithm selected for the training of the network was backpropagation of errors which consisted on the weights update to minimize the network prediction error. The selection of weights was such that the fitting error converged to a predetermined threshold. A total of 35 low correlated initial data points were used for the training of the network. The network was required to identify designs with excellent performance rather than accurately finding the global minimum. Accordingly, a low number of initial data points were selected but it was crucial the samples were space filling to provide an acceptable prediction across the whole design space.

The next step was the minimization of the average temperature on the sleeves of the compressor. The use of global search optimization algorithms increased the

Table 2 Design parameters description and bounds

	h_1 (m)	h_2 (m)	w_1 (m)	w_2 (m)	l (m)	m (m)	a ($^\circ$)
Min	0.066	0.066	0.066	0.066	0.137	0.05	11
Max	0.257	0.257	0.235	0.235	0.161	0.148	11

likelihood of finding the better performing configurations because it increased the portion of the design space that was explored. The optimization problem has been formulated as continuous single objective and unconstrained. Based on the description of the problem, the algorithm selected was a multi-start gradient-based. 10 starting points were selected from the design space by a Latin Hypercube design of experiments with minimization of the correlation to enlarge the portion of the design space explored. The lowest of the 10 local minima found was selected to be the potential global minimum.

The performance of the predicted optimal design was compared to the CFD based calculated performance of the same design. Next, the computational performance value was included into the database to increase the prediction resolution of the surrogate model in that area of the design space. Then, a new surrogate model was trained and optimized. Consequently, the surrogate based optimization process was an iterative practice presented in the flow diagram of the whole optimization process depicted in Fig. 2.

The optimization loop ran until the convergence criteria was met. In this case, two convergence criteria were considered: the first one was the accomplishment of a new design that performed significantly better than the baseline design and better than any design from the initial set of samples. The second one was the wall time allocated to find an optimum solution. The solution found was the optimum since it was the best solution possible found with the constraints of computational resources and time.

Experimental Validation

The experimental validation consists of the physical replication of the CFD model trying to minimise the influence of external components. The behaviour of the compressors in operation has been replicated. The gas generator chassis has been preserved for both the current and optimal designs only changing the modified components. The compressors and fans have been maintained the same for both configurations. The validation consists of the firm attachment of 8 spaced thermocouples to the surface of the critical parts. The position of the thermocouples over the surface of the compressors have remained unchanged in both tests. In addition, the ambient temperature as well as the compartment inlet and outlet temperatures have been monitored. As the objective function for optimization was the average temperature of the critical parts, the experimental average temperature has been calculated as the average temperature on those 8 discrete locations for both current and optimised design and those have been compared.

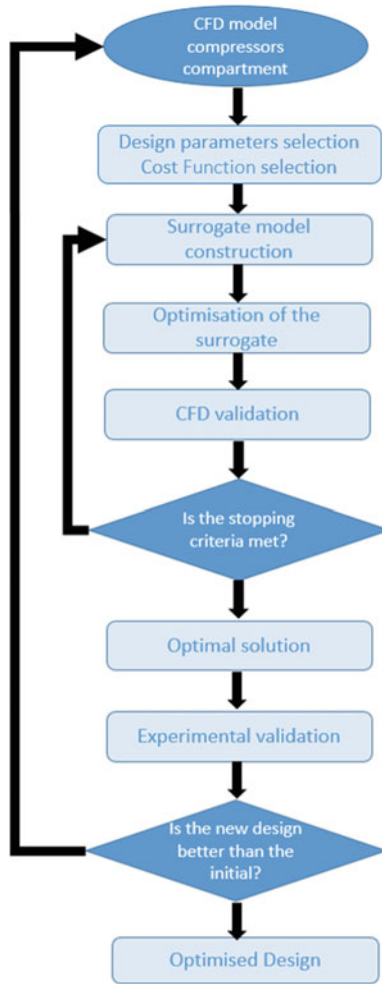


Fig. 2 Flow diagram of the full design optimization process

Results and Discussion

Within the allocated time, a total of 15 optimizations were performed by following the procedure described in the Fig. 2. The parameters obtained for each one of the designs as well as the average and maximum temperature on the surface of the critical parts are depicted in the Table 3.

The performance of each of those designs is better than the baseline design in terms of the average temperature of the sleeves (339.78 K). In this case, the performance must be understood as a high likelihood of physical improvements on the cooling performance rather than a quantification of the temperature decrease. 5 opti-

Table 3 Computational results of the optimization

It.	h_1 (m)	w_1 (m)	h_2 (m)	w_2 (m)	l (m)	m (m)	a (rad)	Av. T. (K)	Max.T (K)
1	0.204	0.066	0.151	0.235	0.161	0.064	6.091	323.44	412.34
2	0.257	0.235	0.144	0.081	0.161	0.062	6.091	320.48	411.12
3	0.221	0.168	0.257	0.235	0.137	0.117	0.192	328.89	459.93
4	0.221	0.235	0.173	0.235	0.137	0.050	6.091	337.25	446.33
5	0.159	0.066	0.085	0.235	0.141	0.050	6.091	325.77	411.58
6	0.196	0.066	0.257	0.235	0.141	0.148	0.192	326.43	421.18
7	0.257	0.235	0.066	0.140	0.161	0.095	0.160	323.62	423.05
8	0.257	0.235	0.257	0.137	0.161	0.050	0.192	317.18	393.37
9	0.230	0.066	0.245	0.235	0.137	0.148	0.192	325.17	394.94
10	0.174	0.124	0.066	0.162	0.161	0.050	0.192	325.14	420.06
11	0.257	0.235	0.066	0.115	0.156	0.050	0.192	319.20	384.21
12	0.220	0.235	0.066	0.235	0.158	0.133	6.091	319.15	392.26
13	0.257	0.235	0.066	0.109	0.137	0.050	0.192	324.55	427.98
14	0.196	0.235	0.123	0.235	0.161	0.050	6.091	322.31	405.76
15	0.209	0.235	0.066	0.235	0.161	0.120	6.091	318.51	375.61

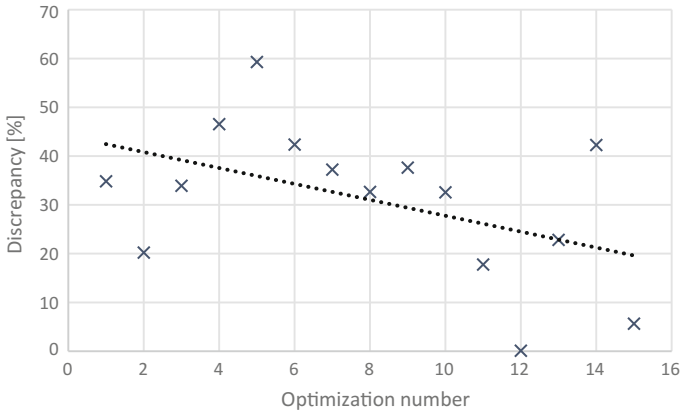


Fig. 3 Comparison surrogate predicted and CFD calculated temperature

mal designs had a better performance than each one of the initial data points used to construct the surrogate model. This fact justifies the investment of effort and time on using a CFD based surrogate based optimization approach to find a better performing design. In addition, the use of a surrogate model was key to enable the design optimization of the compressors compartment within the limited available time. The predicted performance of the optimal designs was compared to their CFD calculated performance. Discrepancies were found between the two which were calculated as the absolute difference between predicted and calculated average temperature on the sleeves. These discrepancies are presented over the number of iterations in the Fig. 3. One can see the discrepancies were decreasing with the number of iterations. This was caused by an increase in resolution of the surrogate model caused by the addition of the optimization samples to build the surrogate model. The results suggest in the case of the number 14, there was not enough information from that area of the design space and the prediction was poor. The objective of the implementation of the optimization methodology was to find designs with a rather improved CFD performance compared to the baseline design. Although there are discrepancies between predicted and computed performances, those are irrelevant because the CFD based performance of the optimal designs found was significantly improved.

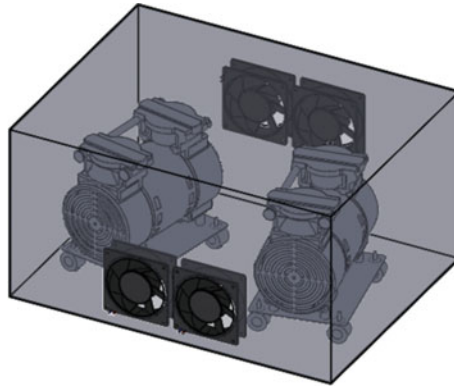
The design corresponding to the 15th optimization iteration was selected for prototyping and validating the approach. The maximum temperature and the individual average and maximum temperature on each sleeve were also considered. The calculated average and maximum temperatures in each individual sleeve for baseline and optimized designs are presented in Tables 4 and 5. One can see that the optimized design attained lower average and maximum temperatures in every single sleeve. In addition, except from the average temperature on the sleeve 1, the rest of the location presented temperature improvements significantly larger than the discretization error induced by the selection of a coarse mesh. Furthermore, the computational average on the sleeves from the optimal design selected, which is presented in Table 3,

Table 4 Computational cooling performance on the sleeves 1 and 2

Design	Sleeve 1 Av. T (K)	Sleeve 1 Max. T (K)	Sleeve 2 Av. T (K)	Sleeve 2 Max. T (K)
Baseline	325.65	406.88	349.51	453.48
Optimized	318.96	366.61	315.88	364.37
Difference	6.69	40.27	33.63	89.11

Table 5 Computational cooling performance on the sleeves 3 and 4

Design	Sleeve 3 Av. T (K)	Sleeve 3 Max. T (K)	Sleeve 4 Av. T (K)	Sleeve 4 Max. T (K)
Baseline	333.85	420.23	340.66	413.48
Optimized	319.02	374.38	320.20	370.56
Difference	14.83	45.85	20.46	42.92

**Fig. 4** Optimal design of the compartment

reduced the average temperature on the sleeves by 21.27 K which is greater than the discretization error as well (4.29 K). These results suggested a very high likelihood of physical cooling improvements.

The parameters of the selected optimal design were embedded into the drawings of the chassis of the gas generator for prototyping. The CAD model of the optimal design is presented in the Fig. 4.

The experimental validation was conducted as it is described in the Section “[Experimental Validation](#)”. Assuming the ambient temperature was the same for both experimental tests on the current and optimized design and the temperature of the inlet cooling air was maintained as well, the experimental average temperature on the sleeves is presented in the Table 6 and compared to the computational results presented before.

The experimental results differ from the cooling improvements predicted by CFD calculations. The cause of this mismatch was identified as heat generation in the

Table 6 Experimental and computational average sleeves average temperature

Design	Experimental Av. T (K)	Computational Av. T (K)
Baseline	339.93	339.78
Optimized	338.23	318.51

sleeves not being constant and dependent on the overall cooling of the compressors. This would require implementing feedback boundary conditions on the sleeves and that would require further experimental testing to understand and quantify the coupling. Since the construction of the CFD model was based in a single design, the understanding of the behavior and the complexity of the system was limited.

Conclusions

The methodology above presented for the design optimization of fluid systems can be defined as promising since it found a design with outstanding CFD calculated performance compared to the initial design on the basis of the definition of the problem and the objective function selected. The goal of the work was to implement a robust design optimization strategy for deployment in an industrial design environment and at this stage its accomplishment requires further work. The physics of the problem selected for that purpose which was the cooling of the sleeves of the compressors, was challenging and unexpectedly complex. The sleeves were identified to have a dynamic rate of heat generation dependent on the cooling of the motors of the compressors and the cooling was also linked to heat conduction across the walls of the enclosure. For that reason, the modelling of the heat sources as independent and constant may have been responsible for the divergence of the real world and the CFD results which forecast a significant temperature decrease on the critical parts of the compressors.

To robustly model those complex problems, it is mandatory to build a CFD model based on more than one real-world designs when in this case it was constructed based on only one design. From an industrial point of view, the time experimental resources and computational resources required to address this problem limit the routine application of this strategy for problems with either complex geometry or physics. However, there is a clear opportunity for the implementation of this methodology in simpler problems which are better understood and more easily modelled. The approach can deliver well performing designs on the basis of the definition of the problem. Therefore, if the CFD model represents the reality the potential improvements predicted will be realized.

Further work will be on the implementation of this strategy for the optimization of simpler components which can be modelled and robustly validated and the time required to obtain a solution is significantly shorter than for the optimization of the

cooling of the compressors compartment. This will accomplish the goal of delivering a design optimization strategy for an industrial design background.

In terms of the optimization of the cooling of the compressors compartment, further work should be conducted on the quantification of the dynamic boundary conditions based on further testing. It would be beneficial to test more than one to fully understand and quantify the thermal behavior of the compressors.

Acknowledgements The authors would like to acknowledge the funding support provided by Peak Scientific Instruments Ltd, the University of Strathclyde and the Knowledge Transfer Partnership(KTP) Scheme of Innovate UK.

[This work is part of the Knowledge Transfer Partnership project number 10025 between the University of Strathclyde and Peak Scientific Instruments Ltd.]

References

- Chen S, Liu Y (2002) An optimum spacing problem for three-by-three heated elements mounted on a substrate. *Heat Mass Transf* 39(1):3–9
- Dias T, Milanez LF (2006) Optimal location of heat sources on a vertical wall with natural convection through genetic algorithms. *Int J Heat Mass Transf* 49(13):2090–2096
- Hotta TK, Venkateshan SP (2015) Optimal distribution of discrete heat sources under natural convection using ann–ga based technique. *Heat Transfer Eng* 36(2):200–211
- Kadiyala PK, Chattopadhyay H (2011) Optimal location of three heat sources on the wall of a square cavity using genetic algorithms integrated with artificial neural networks. *Int Commun Heat Mass Transfer* 38(5):620–624
- Madadi RR, Balaji C (2008) Optimization of the location of multiple discrete heat sources in a ventilated cavity using artificial neural networks and micro genetic algorithm. *Int J Heat Mass Transf* 51(9):2299–2312
- Product data sheet 9956—*EBM-Papst*
- Soleimani S, Ganji D, Gorji M, Bararnia H, Ghasemi E (2011) Optimal location of a pair heat source-sink in an enclosed square cavity with natural convection through PSO algorithm. *Int Commun Heat Mass Transfer* 38(5):652–658
- SolidWorks flow Simulation (2016) technical reference. Dassault Systemes
- Sudhakar T, Balaji C, Venkateshan S (2009) Optimal configuration of discrete heat sources in a vertical duct under conjugate mixed convection using artificial neural networks. *Int J Therm Sci* 48(5):881–890
- Sudhakar T, Shori A, Balaji C, Venkateshan S (2010) Optimal heat distribution among discrete protruding heat sources in a vertical duct: a combined numerical and experimental study. *J Heat Transfer* 132(1):011401

Delaunay-Based Global Optimization in Nonconvex Domains Defined by Hidden Constraints



Shahrouz Ryan Alimo, Pooriya Beyhaghi and Thomas R. Bewley

Abstract This paper introduces a new surrogate-based optimization algorithm to optimize a deterministic objective function with non-computable constraint functions (a.k.a. hidden constraints). Both the objective function and the feasible domain are defined within a known rectangular domain. The objective function might be nonconvex, computationally expensive, and without analytic expression. Moreover, the feasible domain boundaries are not explicitly defined, but can be determined via oracle calls (feasible or not) and learned as the algorithm proceeds. To solve this class of optimization problems, the proposed algorithm, in each iteration, approximates the feasible domain boundary by incorporating a Support Vector Machine (SVM) classifier model as an approximation for the non-computable constraint function, which characterizes the feasible domain. The uncertainty associated with this surrogate is modeled using an artificially-generated uncertainty function built on the framework of Delaunay triangulation. This work extends the Delaunay-based optimization algorithm with nonconvex constraints, dubbed Δ -DOGS(Ω), and extends this approach to estimate the feasible domain with binary oracle calls. Similarly, this algorithm at each iteration determines a minimizer of the objective function surrogate model with the highest probability of being feasible. We evaluate the performance of the algorithm through the numerical experiments on a representative test problem.

S. R. Alimo (✉) · T. R. Bewley
UC San Diego, San Diego, CA, USA
e-mail: shahrouz.ryan.alimo@gmail.com

T. R. Bewley
e-mail: bewley@ucsd.edu

P. Beyhaghi
ASML-Cymer San Diego, San Diego, CA, USA
e-mail: p.beyhaghi@gmail.com

Introduction

This paper aims at solving the optimization problems of the form

$$\begin{aligned} \text{minimize } f(x) \quad \text{with } x \in \Omega := L_c \cap L_s \subseteq \mathbb{R}^n \quad \text{where} \\ L_c = \{x | c(x) \leq 0\}, \quad L_s = \{x | a \leq x \leq b\}, \end{aligned} \quad (1)$$

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, and Ω is defined with hidden constraint functions. The only available measurements are of the sign $\{c(x)\}$. Then $c(x)$ is treated as binary measurement (e.g., ignoring information about the distance to the boundary feasible region); therefore measurements indicate “feasible” and “infeasible” regions Lee et al. (2011).

Motivation

Having a feasible domain that is not defined explicitly is common in some industrial applications such as shape optimization Gramacy et al. (2016) and chemical reactions Gelbart et al. (2018). In these situations, the optimums of the objective function must be further evaluated, and the solutions can be acceptable or unacceptable. That is, the feasible domain is only available through costly oracle calls. Nevertheless, in working conditions, these applications need to be tuned via a limited set of adjustable continuous parameters; for instance, in shape optimization, problems can be solved using only a handful of adjustable parameters usually modeled with $n < 10$. Most of the trial-and-error approaches (e.g) to do so could be both computationally expensive and time consuming. Thus, there is an ever-increasing need to develop efficient frameworks that could limit the number of adjustments needed and to automate the work of these complex systems.

The objective of this work is to develop a new optimization method which is designed for when the objective function is expensive to calculate. In addition, the constraint violation comes from the same process as the objective function, but is accessible through binary oracle calls (acceptable or unacceptable). In these settings, the response of the simulator/system is whether a constraint is violated or not, and no further information about the simulator is given Lee et al. (2011). Thus, the proposed method needs to simultaneously estimate the feasible region (constraint region) and solve the minimization problem.

Many modern optimization approaches for shape optimization of computer-aided designs converge without derivative information and require only weak regularity conditions Gramacy et al. (2016), Alimo et al. (2017), Marsden et al. (2004), Moghadam et al. (2012). When the constraint function is hidden, a simulation displays a flag indicating whether a capacitor has been become full during the simulation, but we know neither when this occurred nor the level of capacitor charge. Such

problems are some of the most challenging optimization problems, and most of the existing approaches rely on heuristic approaches Digabel and Wild (2015).

One of the first schemes for solving such a family of problems was introduced by Conn et al. (1998), where they considered the trust-region subproblem. The authors also described virtual constraints as non-computable constraints and recommend using an extreme-barrier approach. This approach is based on restricting the solution to inside the trust-region subproblem, and based on whether the point is feasible or not, the trust region is adjusted. Further improvement is made by using an extreme-barrier approach.

Regarding other approaches, Gramacy et al. (2016) developed a statistical approach based on Gaussian processes and Bayesian learning to both approximate the unknown function and estimate the probability of meeting the constraints. In another approach, Lee et al. (2011) forced the problem (1) into an existing statistical framework by using treed Gaussian processes for response surface prediction, and random forests for constraint violation prediction. Finally, there are some recently introduced algorithms Picheny et al. (2016), Gelbart et al. (2018) that are based on Augmented Lagrange, and they try to improve the performance of such schemes. However, most of these algorithms are statistical-based methods, and they usually consider the underlying signal as a realization from a random process. Moreover, most of these algorithms are not globally convergent, and they only have the potential to search globally.

A Delaunay-based optimization algorithm, Δ -DOGS, was a recently developed derivative-free optimization algorithm. This algorithm was extended in Δ -DOGS(Ω) in order to solve, with remarkable efficiency, optimization problems with nonconvex and computationally expensive objective and constraint functions Alimo et al. (2017), Alimo et al. (2018). This new algorithm is provably convergent under the appropriate assumptions. This algorithm can be classified as a response surface method which iteratively solves a subproblem based on interpolations not only of the objective and constraint functions over existing datapoints, but also a synthetic model of the uncertainty of these interpolants, which itself is built on the framework of a Delaunay triangulation over existing datapoints. Unlike other response surface methods, this algorithm can employ any well-behaved interpolation strategy.

This paper introduces a new scheme based on Δ -DOGS(Ω) that solves problems where the feasible domain is not known and can only be approximated using a number of oracle calls within the parameter space. That is, this work solves problems where the constraint functions are hidden functions as (2).

The remainder of the paper is organized as follows. Section 2 briefly reviews the optimization algorithm Δ -DOGS(Ω) that was proposed for solving problems with nonconvex and computationally expensive constraint functions. Section 3 describes the extension of this method to solve problems with hidden constraints. Section 4 illustrates the behaviour of the new method on a representative test problem. Some conclusions are made in Sect. 5.

Review of Δ -DOGS(Ω)

Delaunay-based optimization is a generalizable family of practical, efficient, and provably-convergent derivative-free algorithms designed for a range of nonconvex optimization problems with expensive function evaluations Beyhaghi et al. (2015), Beyhaghi and Bewley (2016). This framework, dubbed Δ -DOGS, was extended by an algorithm Δ -DOGS(Ω) in order to solve optimization problems with both nonconvex and computationally expensive objective and constraint functions Alimo et al. (2017), Alimo et al. (2018). Δ -DOGS(Ω) solves problems in the form of:

$$\begin{aligned} \text{minimize } f(x) \quad \text{with } x \in \Omega := L_c \cap L_s \subseteq \mathbb{R}^n \quad \text{where} \\ L_c = \{x | c_\ell(x) \leq 0\}, \quad L_s = \{x | a \leq x \leq b\}, \end{aligned} \tag{2}$$

where both $f(x)$ and $c_\ell(x)$ for $\ell = 1, \dots, m$ are twice differentiable and possibly non-convex functions which map $\mathbb{R}^n \rightarrow \mathbb{R}$ within the search domain L_s . The optimization problem (2) has two sets of constraints:

- (a) a set of $2n$ bound constraints that characterize the n -dimensional box domain $L_s = \{x | a \leq x \leq b\}$, dubbed the *search domain*, and
- (b) a set of m possibly nonlinear inequality constraints $c_\ell(x) \leq 0$ that together characterize the possibly nonconvex domain L_c , dubbed the *constraint domain*.

The *feasible domain* is the intersection of these two domains, $\Omega := L_s \cap L_c$.

In many application based problems, we are seeking a feasible point $x \in \Omega$ such that $f(x) \leq f_0$, where f_0 is a target value. In this work, we assume that there is a known target value f_0 , which is achievable; i.e., $\exists x \in L_s$ such that $f(x) \leq f_0$ and $c_\ell(x) \leq 0$ for all $\ell = 1, \dots, m$. The $c_\ell(x)$ are nonlinear, computationally expensive constraint functions. However, it is worth noting that the present algorithm can be easily extended to problems for which a target value and constraint violation thresholds are not available, as in Algorithm 1 of Alimo et al. (2018).

We assume that $f(x)$ and $c_\ell(x)$ are computable everywhere in L_s computationally expensive, and possibly nonconvex. For the purpose of the scheme development, we assume that $f(x)$ and $c_\ell(x)$ are twice differentiable. Also, the gradient information for $f(x)$, or its estimate, is usually not available. Moreover, we consider the optimization problems with few adjustable parameters $n \leq 10$. Before presenting the algorithm, we introduce some preliminary concepts:

Definition 1 Given S^k as a set of points that includes the vertices of domain L_s at iteration k of Δ -DOGS(Ω), we define $p^k(x)$ and $g_1^k(x), \dots, g_m^k(x)$ as a set of successive interpolations for the objective and constraint functions $f(x)$ and $c_1(x), \dots, c_m(x)$, respectively, at iteration k . Consider

$$T^k(x) = \max \left[p^k(x) - f_0, g_1^k(x), \dots, g_m^k(x) \right], \tag{3}$$

then the **continuous search function** is defined as

$$s_c^k(x) = \begin{cases} T^k(x)/e^k(x), & \text{if } T^k(x) \geq 0, \\ T^k(x), & \text{otherwise.} \end{cases} \tag{4}$$

Δ -DOGS(Ω) estimates (2) with a set of surrogates, and in the situation where there exists a target value for the objective function such that $f_0 \leq f(x)$ and there are m different nonconvex constraint functions defining the feasible domain, then the algorithm iteratively evaluates the minimizer of (4). This method is an iterative algorithm, and at each iteration, it generates a set of points to estimate the objective functions and the feasible domain. Using the set of available datapoints, the objective functions are approximated with interpolation/regression models, and the uncertainty of the interpolants is quantified using an artificially generated uncertainty model based on the Delaunay triangulation framework. The most expensive part in this calculation is the function evaluation process.

Algorithm 1 Δ -DOGS (Ω) for accurate objective and constraint function evaluations Alimo et al. (2018).

1. Set $k = 0$. Take the set of initialization points S_0 as all vertices of the feasible domain Ω
 2. Calculate (or, for $k > 0$, update) an appropriate interpolating functions $p(x), g_\ell(x)$ through all points in S_k
 3. Calculate (or, for $k > 0$, update) a Delaunay triangulation Δ^k over all of the points in S_k
 4. Find $x_k \in \Omega$ as a global minimizer of $s(x)$ subject to $CS_\ell(x) \leq 0$, for $\ell = 1, 2, \dots, m$.
 5. Calculate $f(x), c_\ell(x)$ at x_k , and take $S_{k+1} = S_k \cup x_k$.
-

In convergence analyses for Δ -DOGS(Ω), the objective and constraint functions were assumed to be twice differentiable with the search domain L_S .

Furthermore, we can observe for Δ -DOGS(Ω), Algorithm 1, that

- if $f_0 \geq f(x^*)$, Δ -DOGS (Ω) generates an infinite sequence of points whose limit points are characterized by objective function values less or equal to f_0 , or
- if $f_0 < f(x^*)$, Δ -DOGS (Ω) generates an infinite sequence of points which is dense everywhere in the feasible domain.

Remark 1 With existence of a target value f_0 for the objective function, Algorithm can find a point $x \in \Omega$ which is feasible and $f(x) \leq f_0$, if such point exists. In fact, any limit point of S is feasible, and its objective function is less than f_0 . If there is no point in which $f(x) \leq f_0$ and x is feasible, Algorithm will go dense everywhere in the feasible domain Ω .

In the following section we extend Δ -DOGS (Ω) for the cases where the evaluation of the constraint function $c(x)$ is only limited to its sign as shown in Fig. 1.

Unlike the previous work Alimo et al. (2018) for Δ -DOGS(Ω) where the function evaluation was computable, in the next section, the constraint violation is considered

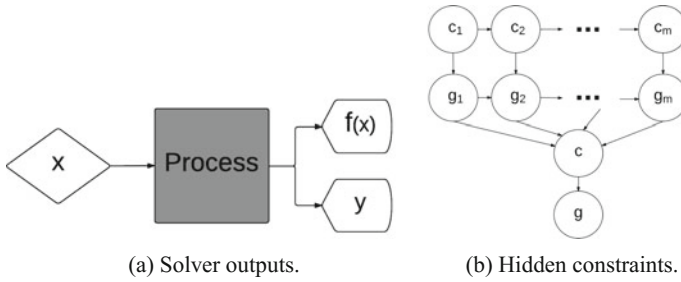


Fig. 1 Function evaluation process for the problems in form of (1). **a** Each constraint violation is measured only with feasible or infeasible as $y_i = \text{sign}\{c(x^{(i)})\}$. **b** The feasible domain is modeled using a surrogate model g

non-computable and is only limited to an oracle call or a binary measurement (feasible or infeasible). In this problem, we are not able to measure the constraint function, and we are limited to finding the sign of c_ℓ for $\ell = 1, \dots, m$, and the point $x^{(i)}$ is either feasible or infeasible.

Quantifying Constraint Violation Using SVM

In this section, we extend the Δ -DOGS(Ω) algorithm to solve the optimization problem with the feasible domain that is characterized with binary oracle calls (feasible or not). For each set of parameters the constraint violation is determined by

$$y(x_i) = \begin{cases} -1, & \text{if } x_i \text{ is feasible or } c(x) \leq 0 \\ +1, & \text{if } x_i \text{ is infeasible or } c(x) > 0. \end{cases} \tag{5}$$

We assume that the information about the feasible domain boundaries are only available through function (5). In other words, since the constrains are not defined explicitly, we only can determine if a point of interest is within the feasible domain or not. This situation is similar to the binary classification problems where the training points (data) are divided into two classes. The final goal of a classification problem is to predict the class of a new candidate point. In this work, we borrow that idea and extend it to use in the optimization problem (2) to estimate the boundary of feasibility as the algorithm proceeds.

There are a wide variety of classifiers in the machine learning literature such as Perceptron, Artificial Neural Network (ANN), and Support Vector Machine (SVM).

Perceptron is a linear classifier that tries to find a hyperplane that separates the labeled training data points into two classes so that points with the same label stays in one side of the hyperplane. The Perceptron algorithm starts with an initial hyperplane. At each iteration of the algorithm it processes a point from the training set and

update the weight of parameters to characterize the optimal hyperplane classifier. The algorithm is suitable for online learning problems where the training data points are given sequentially. In addition, computationally it is very appealing. However, it requires a relatively large training set and the final hyperplane is not necessarily the best hyperplane that separates the two classes Cortes and Vapnik (1995). Moreover, since this problem is not linearly separable using Perceptron is not practical.

ANN Goodfellow et al. (2016) impose multiple-layer linear classifiers that can classify data points using only a handful of features, but they include many undetermined and hidden layers. The drawback for this approach is that there are many tuning parameters to characterize these hidden layers correctly. These tuning parameters are problem specific, and they require a large amount of data to train the hidden layers of the network Bengio et al. (2015).

SVM Cortes and Vapnik (1995), Cherkassky and Ma (2004) is a linear classifier, which aims at determining the maximum margin hyperplane. SVM classifier transfers features into a higher dimensional space using appropriate kernels and then fits a linear classifier. As a result it can also separate the data points that are not linearly separable. Overall, the SVM classifier is computed by solving a quadratic programming optimization problem.

At each iteration, the Δ -DOGS(Ω_H) search for a minimizer of the optimization problem which has the highest probability of being inside the feasible domain. The oracle calls are assumed to be computationally expensive therefor the adapted classifier should be able to estimate the boundaries using fewer samples. In this paper, we propose a SVM-based approach to approximate the feasible domain of solutions at each iteration of the optimization algorithm.

We assume that the boundaries of the feasible domain are twice differentiable with a bounded Lipschitz norm. In other words, there exists an unknown underlying function in which $\text{sign}\{c(x)\} = \text{sign}\{g(x)\}$ for all $x \in L_s$. To be able to approximate a wide variety of boundary functions, we consider the Radial Basis Functions (RBF) as the kernel of the SVM. That is

$$g(x) = \sum_{i=1}^d w_i \phi_i(x) + b, \tag{6}$$

where we consider b as a bias term and $\phi(x)$ as radial basis kernel functions that denote the feature space transformation. Let $y_i = \text{sign}\{g(x_i)\}$, then the distance of a point x_i to the decision surface $g(x_i)$ is given by

$$\frac{y_i g(x_i)}{\|w\|} = \frac{y_i (w^T \phi(x) + b)}{\|w\|} > 0.$$

Note that x_i for $i = 1, \dots, N$ are evaluated points in the feature (parametric) space at the Nth iteration. In this way, every point x_i can be transformed into a d -D space such that $X_i = [\phi_1(x_i), \phi_2(x_i), \dots, \phi_d(x_i)]$.

The optimal hyperplane is determined by solving a quadratic programming (QP) as follows:

$$\min_{z \in \mathbb{R}^{d+1}} z^T L z, \quad \text{subject to } Y \cdot (A z) \geq 1, \quad \text{where}$$

$$L = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} F & 1^T \\ 1 & 0 \end{bmatrix},$$

$$Y = \begin{bmatrix} \text{sign}\{c(x_1)\} \\ \vdots \\ \text{sign}\{c(x_N)\} \\ 0 \end{bmatrix}, \quad z = \begin{bmatrix} w_1 \\ \vdots \\ w_d \\ b \end{bmatrix}.$$

where $F_{i,j} = \phi_i(x_j) = \varphi(|x_i - x_j|)$ and $1 = ([1, \dots, 1]^T)_{N \times 1}$. Performance depends on the choice of basis functions $\phi(x)$ used to leverage the evaluated dataset, and the choice of SVM kernels are problem-dependent. We set

$$\phi_i(x) = \varphi(r) \quad , \text{ where } r = |x_i - x|$$

The most well-known RBF models are the Gaussian kernel model $\varphi(r) = e^{-r^2/\sigma^2}$, the polynomial model $\varphi(r) = r^3$, and the inverse multi-quadratic kernel model $\varphi(r) = 1/\sqrt{\sigma^2 + r^2}$.

The main challenge in optimization with virtual constraints is that there are many models of g that could successfully classify the two classes from each other, and these approximated constraint functions could have a very different range of values from one another. The variety in appropriate g models stems from the fact that we only have access to the sign of $c(x)$. As a result, the inclusion of more data points from specific regions can sometimes lead to estimated g models deviating from the true hidden function. To control this deviation, we scale the estimated constraint function using the initial training data points in order to have the same range of variations as the objective function $f(x)$.

Δ -DOGS(Ω_H)

The new algorithm is designed based on the Δ -DOGS(Ω). Since the constraints are hidden, an approximation (6) based on the SVM is considered for the model of constraints. The search function is defined as

$$s_c(x) = \max\left\{\frac{p(x) - f_0}{r_f e(x)}, \frac{g(x)}{r_g e(x)}\right\} \tag{7}$$

where $g(x)$ is modeled as (6) and r_f and r_g are constants to make the constraint search function model in the same range of variation as the objective search function.

Δ -DOGS(Ω_H) algorithm depends upon a handful of adjustable parameters, the selection of which affects its rate of convergence. The remainder of this section discusses heuristic strategies to tune these algorithm parameters, noting that this tuning is an application-specific problem, and alternative strategies (based on experiment or intuition) might lead to more rapid convergence for certain problems.

The first task encountered during the setup of the optimization problem is the definition of the design parameters and search domain L_s . Note that the feasible domain considered during the optimization process is characterized by simple upper and lower bounds for each design parameter; normalizing all design parameters to lie between 0 and 1 is often beneficial Alimo et al. (2018), Beyhaghi and Bewley (2018).

The second challenge is to scale the objective function $f(x)$ and the hidden constraint function $g(x)$ themselves, such that the range of the normalized functions $f(x)$ and $g(x)$ over the search domain L_s are the same and about unity.

If an estimate of the actual range of $c(x)$ (the same as $g(x)$) is not available a priori, we may estimate it at any given iteration using the available measurements.

Results

The test function is considered as a quadratic objective function, given by the distance from a point in the search domain L_s , and is defined over an n -dimensional space, subject to a nonlinear inequality constraint. The feasible domain is generated using the sign of a Rastrigin function, defining a disconnected feasible domain characterized by 2^n distinct “islands” within the search domain:

$$\min_{x \in L_s} f(x) = \|x - x_0\|^2 - 0.024n, \tag{8a}$$

$$\text{subject to } \text{sign}\{c(x)\} < 0, \tag{8b}$$

$$c(x) = \frac{n}{12} + \frac{1}{10} \sum_{i=1}^n \left\{ 4(x_i - 0.7)^2 - 2 \cos\left(4\pi(x_i - 0.7)\right) \right\},$$

$$0 \leq x_1, x_2, \dots, x_n \leq 1. \tag{8c}$$

This problem has 2^n local minima, including the unique global minimum where $x_0 = [0.19, 0.29]^T$ for $n = 2$ with $f(x^*) = f(x_0) = 0$.

The results show that Δ -DOGS(Ω_H), the newly introduced method, has the potential to identify the global minimizer of the objective function under the feasible region. It is observed that the choice of kernel function acts as a crucial factor in enabling the global convergence of the optimization method. Figures 2 and 3 show the results using a piece-wise linear kernel. This is an appropriate choice since the underlying model of constraint function range will not change exponentially when the algorithm detects a feasible region, as was the case when a cubic kernel was used. In addition, use of a cubic kernel did not lead to global convergence.

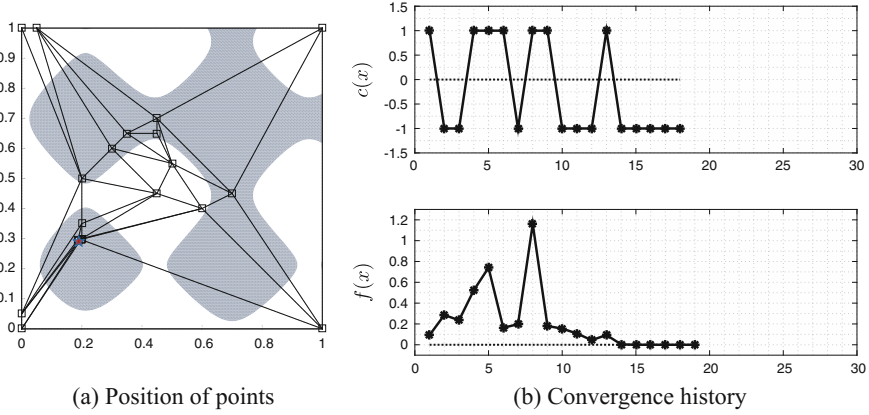


Fig. 2 Nonconvex problem for $n = 2$ with vertices as initial points. The gray region is the feasible domain. The objective function is the distance function from the global minimizer showed by red star $x^* = [0.19, 0.29]^T$

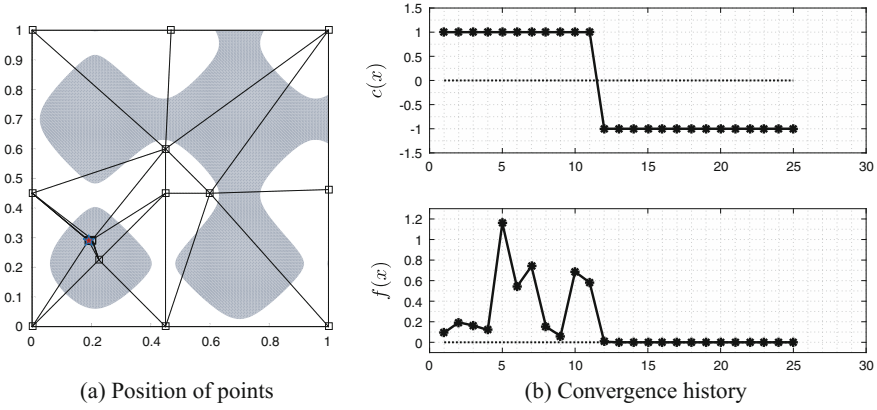


Fig. 3 Nonconvex problem for $n = 2$ with $\{x_{0,i} = 0.35, x_{0,i} + 0.1 e_i\}$ where e_i is the i th main coordinate direction. The gray region is the feasible domain and the objective function is the distance function from $x^* = [0.19, 0.29]^T$

Conclusions

In this work, we presented a new derivative-free algorithm for the optimization of expensive cost functions subject to box constraints and hidden type of constraints, in which a design point can be labeled as feasible or not feasible. We modeled the hidden constraints with the popular classification techniques that is from the machine learning literature. The well known techniques of SVMs are applied in a global optimization framework.

As a future work, we compare the results from Artificial Neural Network and leverage deep learning techniques to quantify $g(x)$. Furthermore, the presented optimization method will be applied to an application-based problem, and we will study the behaviour of $g(x)$ in different problems.

References

- Alimo SR, Beyhaghi P, Bewley TR Delaunay-based derivative-free optimization via global surrogates, part iii: nonconvex constraints. *J Glob Optim*
- Alimo SR, Cavaglieri D, Beyhaghi P, Bewley TR (2017) Design of imexrk time integration schemes via delaunay-based derivative-free optimization. *Journal of Global Optimization*, under preparation
- Alimo S, Beyhaghi P, Meneghello G, Bewley T (2017) Delaunay-based optimization in cfd leveraging multivariate adaptive polyharmonic splines (maps). In: 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 0129
- Bengio Y, Goodfellow IJ, Courville A (2015) Deep learning. *Nature* 521:436–444
- Beyhaghi P, Bewley T (2016) Delaunay-based derivative-free optimization via global surrogates, part ii: convex constraints. *J Glob Optim*, under review
- Beyhaghi P, Bewley T Delaunay-based via lattice-based optimization algorithm. *J Glob Optim*
- Beyhaghi P, Cavaglieri D, Bewley T (2015) Delaunay-based derivative-free optimization via global surrogates, part i: linear constraints. *J Glob Optim* 63:1–52
- Cherkassky V, Ma Y (2004) Practical selection of svm parameters and noise estimation for svm regression. *Neural networks* 17(1):113–126
- Conn A, Scheinberg K, Toint P (1998) A derivative free optimization algorithm in practice. In: 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 4718
- Cortes C, Vapnik V (1995) Machine learning. *Support-vector networks* 20(3):273–297
- Digabel SL, Wild SM (2015) A taxonomy of constraints in simulation-based optimization. arXiv preprint [arXiv:1505.07881](https://arxiv.org/abs/1505.07881)
- Gelbart MA, Snoek J, Adams RP (2014) Bayesian optimization with unknown constraints
- Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press, Cambridge
- Gramacy RB, Gray GA, Le Digabel S, Lee HK, Ranjan P, Wells G, Wild SM (2016) Modeling an augmented lagrangian for blackbox constrained optimization. *Technometrics* 58(1):1–11
- Lee H, Gramacy R, Linkletter C, Gray G (2011) Optimization subject to hidden constraints via statistical emulation. *Pac J Optim* 7(3):467–478
- Marsden AL, Wang M, Dennis JE Jr, Moin P (2004) Optimal aeroacoustic shape design using the surrogate management framework. *Optim Eng* 5(2):235–262
- Moghadam ME, Migliavacca F, Vignon-Clementel IE, Hsia T-Y, Marsden AL (2012) Optimization of shunt placement for the norwood surgery using multi-domain modeling. *J Biomech Eng* 134(5):051002
- Picheny V, Gramacy RB, Wild S, Le Digabel, S (2016) Bayesian optimization under mixed constraints with a slack-variable augmented lagrangian. In: *Advances in neural information processing systems*, pp 1435–1443

Part III
Applications of Optimization in
Engineering Design Automation

Optimized Vehicle Dynamics Virtual Sensing Using Metaheuristic Optimization and Unscented Kalman Filter



Manuel Acosta and Stratis Kanarachos

Abstract This paper presents an Optimized Unscented Kalman Filter for vehicle dynamics virtual sensing. An automated procedure to optimize the virtual sensor parameters based on metaheuristic algorithms is presented in order to avoid the time-consuming and complex manual tuning task. Specifically, Genetic Algorithm Optimization (GA) and contrast-based Fruit Fly optimization (c-FOA) are applied to minimize the estimation error in steady-state and transient driving maneuvers. The virtual sensor is implemented in a high-fidelity vehicle dynamics simulation software (IPG-CarMaker®) and results demonstrate the improvement of the estimation accuracy with respect to a preliminary filter tuning carried out using a systematic trial and error approach.

Introduction

Active Chassis Control Systems seek to enhance the vehicle driving dynamics and keep the vehicle under stable and controllable limits Kanarachos (2013). This task is achieved by monitoring a number of vehicle states which are representative of the vehicle behavior (e.g. lateral velocity, yaw rate, tire forces) Kiencke and Nielsen (2005). As direct measurement of some of these states is still complex and expensive Eom et al. (2014), Matsuzaki et al. (2008), virtual sensing is presented as an attractive alternative to infer the control signals from readily-available measurements.

M. Acosta (✉) · S. Kanarachos
Coventry University, Priory Street, Coventry, UK
e-mail: ac3354@coventry.ac.uk

S. Kanarachos
e-mail: ab8522@coventry.ac.uk

Due to the nonlinear and time-varying characteristics of the vehicle dynamics, Kalman Filtering techniques (Unscented Kalman Filter (*UKF*) and Extended Kalman Filter (*EKF*)) are preferred in the literature to develop the virtual sensors Doumiati et al. (2012), Haudum et al. (2016). Despite the implementation of the Kalman Filter Algorithms in vehicle dynamics is relatively straightforward and has been discussed extensively, little attention has been paid to the cumbersome task of selecting the process and measurement noise covariance matrices (\mathbf{Q} , \mathbf{R}) that minimize the estimation error. These values are often determined empirically, following an iterative trial and error method. The necessity of a systematic and automated tuning methodology for vehicle dynamics applications has been mentioned briefly in a reduced number of works Haudum et al. (2016), Koch (2011), Gadola et al. (2014), but still, a deep analysis regarding the optimization techniques employed for this purpose is missing. The necessity of automating this procedure takes even more importance if covariance-scheduling strategies are envisaged Kiencke and Nielsen (2005). As the uncertainty associated with the vehicle behavior changes drastically with the level of lateral excitation or the transient content of the driving maneuver, several optimized parameters might be required and varied according to the driving situation (e.g. level of lateral acceleration, yaw acceleration or longitudinal velocity Klier et al. (2008), Tuononen (2009)). Regarding other engineering applications, the Kalman Filter tuning has received greater attention, and numerical optimization Kaur and Kaur (2016), Kaur and Kaur (2016), Gadola et al. (2014), Gaussian optimization Scardua and da Cruz (2016), autocovariance least-squares methods Åkesson et al. (2007), reinforcement learning Goodall and El-Sheimy (2007) or metaheuristic optimization Ting et al. (2014) have been suggested to avoid the rudimentary and time-consuming trial and error approach.

In this paper, an automated tuning methodology based on metaheuristic optimization is proposed. An objective function composed of the longitudinal velocity, lateral velocity, and yaw rate weighted errors is minimized using the well-known Genetic Algorithms (*GA*) and a recently-developed contrast-based Fruit Fly Optimization routine Kanarachos et al. (2017) (*c-FOA*). The optimized virtual sensor parameters have been determined for two different driving situations: *steady-state* and *transient* maneuvers. This will facilitate the development of adaptive virtual sensing strategies in future stages of this research. Moreover, the proposed algorithms have been benchmarked against other popular optimization routines (Sequential Quadratic Programming (*SQP*), *Nelder-Mead*, Artificial Bee Colony (*ABC*), Particle Swarm Optimization (*PSO*) and Differential Evolution (*DE*)). The rest of the paper is structured as follows. In Section “[Background](#)”, the *UKF* algorithm and its implementation for vehicle dynamics estimation are introduced. The section is completed with a brief insight into *GA*, *c-FOA*, and the definition of the evaluation function. The optimization procedure is described in Section “[UKF Design and Optimization](#)”, followed by an analysis of the results obtained with the optimized virtual sensors, Section “[Results](#)”. Finally, a brief discussion on the performance of the virtual sensors and further research steps are indicated in Section “[Conclusions](#)”.

Background

Unscented Kalman Filter

UKF is a nonlinear Kalman filter employed when the problem exhibits strong nonlinearities and the linearization of the plant required by other nonlinear filters (e.g. *EKF*) might worsen considerably the estimation accuracy Doumiati et al. (2012). It has been demonstrated as a suitable approach to handle the vehicle nonlinear behavior derived from the tire-road characteristics Antonov et al. (2011), Doumiati et al. (2009). To formulate the filter, a generic nonlinear system is expressed in state-space form,

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k) + \mathbf{w}_k, \mathbf{Y}_{k+1} = \mathbf{h}(\mathbf{X}_{k+1}, \mathbf{U}_k) + \mathbf{v}_k \quad (1)$$

where \mathbf{X} , \mathbf{Y} and \mathbf{U} are the vector of states, outputs, and inputs of the system. The terms \mathbf{w} , \mathbf{v} are the process and measurement noises respectively, and are assumed to be Gaussian, uncorrelated, and zero mean (i.e. $\mathbf{w} \approx N(0, \mathbf{Q}), \mathbf{v} \approx N(0, \mathbf{R})$). \mathbf{Q} and \mathbf{R} are referred as the filter tuning covariance matrices.

The filter relies on the Unscented Transformation (*UT*), which consists on the propagation of a small set of deterministically selected sigma-points through the system. After this propagation, the system nonlinearities are inferred from the statistics of these points. The spread of the sigma-points is determined by the scaling parameter λ , which depends on the constants α and κ , related by expression ($\lambda = \alpha^2(L + \kappa) - L$) Rhudy and Gu (2013), Wan and Van Der Merwe (2000), where the length of the state vector is denoted by L . In this paper, the plant and measurement noises are considered additive, and thus the formulation of the *UKF* is reduced to the formulation of the standard or unaugmented *UKF* Rhudy and Gu (2013). The matrix of sigma-points is formed using the Eq. (2), where the number of rows is given by L and the number of columns corresponds to $2L + 1$,

$$\boldsymbol{\chi}_k = \left[\hat{\mathbf{X}}_{k|k}, \hat{\mathbf{X}}_{k|k} + \sqrt{\Theta \mathbf{P}_{k|k}}, \hat{\mathbf{X}}_{k|k} - \sqrt{\Theta \mathbf{P}_{k|k}} \right] \quad (2)$$

where \mathbf{P} is the covariance matrix of the filter and Θ is equal to $(\lambda + L)$. The matrix square root ($\sqrt{\mathbf{P}_{k|k}}$) is calculated using the Cholesky method, which calculates a lower triangular matrix representative of the square root, expression (3).

$$\sqrt{\mathbf{P}_{k|k}} \sqrt{\mathbf{P}_{k|k}}^T = \mathbf{P}_{k|k} \quad (3)$$

The propagation of the sigma points through the nonlinear system is performed using Eq. (4).

$$\boldsymbol{\chi}_{k+1|k}^i = \mathbf{f}(\boldsymbol{\chi}_k^i, \mathbf{U}_k), \quad i \in \{0, \dots, 2L\} \quad (4)$$

After that, the post-transformation mean and covariance values are calculated using weighted averages, (5–6).

$$\hat{\mathbf{X}}_{k+1|k} = \sum_{i=0}^{2L} \eta_i^m \boldsymbol{\chi}_{k+1|k}^i \quad (5)$$

$$\begin{aligned} \mathbf{P}_{k+1|k} &= \mathbf{Q}_k + \\ &+ \sum_{i=0}^{2L} \eta_i^c (\boldsymbol{\chi}_{k+1|k}^i - \hat{\mathbf{X}}_{k+1|k})(\boldsymbol{\chi}_{k+1|k}^i - \hat{\mathbf{X}}_{k+1|k})^T \end{aligned} \quad (6)$$

The weights η_i^c and η_i^m are computed from Eqs. (7–9),

$$\eta_0^m = \frac{\lambda}{\lambda + L} \quad (7)$$

$$\eta_0^c = \eta_0^m + 1 - \alpha^2 + \beta \quad (8)$$

$$\eta_i^c = \eta_i^m = \frac{1}{2(L + \lambda)}, \quad i \in \{1, \dots, 2L\} \quad (9)$$

The parameter β is known as the secondary scaling parameter Rihdy and Gu (2013). Similarly, the matrix of sigma-points is propagated through the observation function (**h**) using expression (10).

$$\boldsymbol{\xi}_{k+1|k}^i = \mathbf{h}(\boldsymbol{\chi}_{k+1|k}^i, \mathbf{U}_k), \quad i \in \{0, \dots, 2L\} \quad (10)$$

The predicted output ($\hat{\mathbf{Y}}_{k+1|k}$), output covariance matrix (\mathbf{P}_{k+1}^{yy}) and cross-covariance matrix (\mathbf{P}_{k+1}^{xy}) are calculated using Eqs. (11–13).

$$\hat{\mathbf{Y}}_{k+1|k} = \sum_{i=0}^{2L} \eta_i^m \boldsymbol{\xi}_{k+1|k}^i \quad (11)$$

$$\begin{aligned} \mathbf{P}_{k+1}^{yy} &= \mathbf{R}_k + \\ &+ \sum_{i=0}^{2L} \eta_i^c (\boldsymbol{\xi}_{k+1|k}^i - \hat{\mathbf{Y}}_{k+1|k})(\boldsymbol{\xi}_{k+1|k}^i - \hat{\mathbf{Y}}_{k+1|k})^T \end{aligned} \quad (12)$$

$$\mathbf{P}_{k+1}^{xy} = \sum_{i=0}^{2L} \eta_i^c (\boldsymbol{\chi}_{k+1|k}^i - \hat{\mathbf{X}}_{k+1|k})(\boldsymbol{\xi}_{k+1|k}^i - \hat{\mathbf{Y}}_{k+1|k})^T \quad (13)$$

The covariance matrices calculated in the previous step are then used to compute the Kalman gain (\mathbf{K}_{k+1}), Eq. (14).

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{\text{xy}} (\mathbf{P}_{k+1}^{\text{yy}})^{-1} \quad (14)$$

Finally, the states estimated in the first stage of the filter are corrected using the expression (15), and the covariance matrix is updated with Eq. (16).

$$\hat{\mathbf{X}}_{k+1|k+1} = \hat{\mathbf{X}}_{k+1|k} + \mathbf{K}_{k+1} (\mathbf{Y}_{k+1} - \hat{\mathbf{Y}}_{k+1|k}) \quad (15)$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1} \mathbf{P}_{k+1}^{\text{yy}} \mathbf{K}_{k+1}^T \quad (16)$$

Application to Vehicle Dynamics Estimation

In this paper the vehicle planar dynamics are approximated using a single-track model Acosta and Kanarachos (2016), Hrgetic et al. (2014). The virtual sensor is completed with a Neural Network (NN) structure, which is aimed to estimate the axle lateral forces required by the UKF using a data-based approach Acosta and Kanarachos (2016). The vehicle planar dynamics Eqs. (17–19) are presented in discrete form as:

$$\begin{aligned} \dot{\psi}_{k+1} &= \dot{\psi}_k \\ &+ \frac{T_s}{I_z} ((F_{yf,k} \cos(\delta_k) + F_{xf,k} \sin(\delta_k))l_f - F_{yr,k}l_r) \end{aligned} \quad (17)$$

$$\begin{aligned} v_{x,k+1} &= v_{x,k} + T_s \dot{\psi}_k v_{y,k} \\ &+ \frac{T_s}{m} (F_{xf,k} \cos(\delta_k) - F_{yf,k} \sin(\delta_k) + F_{xr,k}) \end{aligned} \quad (18)$$

$$\begin{aligned} v_{y,k+1} &= v_{y,k} - T_s \dot{\psi}_k v_{x,k} \\ &+ \frac{T_s}{m} (F_{yf,k} \cos(\delta_k) + F_{xf,k} \sin(\delta_k) + F_{yr,k}) \end{aligned} \quad (19)$$

where T_s is the discretization time employed in the first order approximation, the vehicle mass is denoted by m and the yaw inertia by I_z . The distances from the center of gravity to the front and rear axles are designated by l_f , l_r respectively. Following the approach adopted in other works on state estimation Hrgetic et al. (2014), Doumiati et al. (2012, 2009), the effect of the suspension dynamics on the chassis planar motion is disregarded, and the vehicle planar dynamics are assumed to be entirely represented by the yaw rate, longitudinal velocity, and lateral velocity ($X = \{\dot{\psi}, v_x, v_y\}$). The vector of inputs U is formed by the steering wheel angle and the axle longitudinal forces ($U = \{\delta, F_{xf}, F_{xr}\}$). For the sake of clarity, a reduced filter structure is chosen to introduce the automated tuning problem, and it is assumed that the axle longitudinal forces are estimated in an external block (e.g. with a modular filter structure Acosta et al. (2017)). Finally, the yaw rate and the longitudinal

velocity are the measured quantities ($Y = \{\dot{\psi}, v_x\}$). The axle lateral forces required to compute Eqs. (17–19) at each time step are approximated using a data-based approach, where the relation between these forces (F_y), the axle lateral slips (ϕ), and the longitudinal acceleration (a_x) is presented in the form of a nonlinear function $F_y = f_y(\phi, a_x)$. The axle slips are obtained from the vehicle planar states using a small-angle approximation Kanarachos (2014) Eq. (20).

$$\phi_f = \delta - \left(\frac{\dot{\psi} l_f + v_y}{v_x} \right), \phi_r = - \left(\frac{-\dot{\psi} l_r + v_y}{v_x} \right) \quad (20)$$

The longitudinal acceleration was introduced in the nonlinear function (f_y) in order to take into account the reduction in the available lateral force during combined (braking and cornering) efforts. *NN* were chosen to approximate the tire nonlinear behaviour due to their remarkable fitting capabilities. Specifically, a one-hidden-layer structure (1-10-1) was chosen after a sensitivity analysis and the *NN* were trained employing a *Backpropagation* algorithm and a dataset division of 70/15/15. The training datasets were generated in *IPG CarMaker*® using an experimentally validated vehicle model and a *MF6.1 Pacejka* (2012) tire model. Additional details about this process can be found in Acosta and Kanarachos (2016).

Genetic Algorithms

Genetic Algorithms (*GA*) are global search optimization approaches inspired by the natural biological evolution. These operate based on the principle of survival of the fittest individual among an initial population. The generation of better individuals that are more adapted to the new environment is pursued during the consecutive iterations Chipperfield et al. (2009). Depending on the problem domain, *GA* can be represented using single-level binary strings, integer-coded representations, and real-valued representations. Once the problem representation has been selected, the population size is set and initialized using a stochastic approach. The *GA* routine is then executed under the following operations:

- Selection: Individuals with higher fitness are selected and passed to the new generation based on the theory of “survival of the fittest”.
- Crossover: Off-springs are produced at each new generation using the genetic material of their parents. After several generations, the population tends to orient towards the minimum of the objective function.
- Mutation: A certain biological mutation is introduced in each generation in order to maintain the diversity of the population, thus avoiding early convergence into a local minimum.

Further details regarding *GA* routines are omitted due to space limitations. These can be consulted in Chipperfield et al. (2009), Haupt and Haupt (2004).

Fruit Fly Optimization

Fruit flies are very efficient in detecting food as they can locate it, even if this is 40 km away. Pan, inspired by fruit flies foraging behavior, proposed for the first time a Fruit Fly Optimization Algorithm (*FOA*) on this basis Pan (2012). The authors have developed an enhanced version of the original *FOA* algorithm consisting of the following steps: Swarm generation, Fruit Fly localization, Smell concentration calculation, Best member identification, Current average location selection and Decision delay phase. Additional details regarding this optimization routine are omitted in this paper due to space limitations and can be consulted in Kanarachos et al. (2017).

Objective Function Formulation

In this paper, the fitness function (22) is defined as the sum of the normalized root mean square (*NRMS*) errors (21) of the yaw rate ($e_{\dot{\psi}}$), lateral velocity (e_{v_y}), and longitudinal velocity (e_{v_x}),

$$e = \sqrt{\frac{\sum_{k=1}^N (\hat{X}_k - X_k)^2}{N}} \frac{1}{\max(|X|)} \quad (21)$$

where \hat{X}_k is the vector of estimated states of the *UKF* at time k and X_k is the vector of “true” states calculated in *IPG CarMaker*®.

$$f = e_{v_x} + e_{v_y} + e_{\dot{\psi}} \quad (22)$$

Thus, the *UKF* automated tuning is formulated as an optimization problem consisting of finding the set of *UKF* parameters $\Omega^* = \{Q^*, R^*, \alpha^*\}$ (process covariance matrix Q , measurement covariance matrix R , and *UKF* scaling parameter α), that minimize the objective function f subjected to the boundary (24–25) or integer constraints (26). These constraints depend on the problem representation (continuous or discrete search space), being UB , LB the upper and lower bounds imposed to the continuous search space, and S the discrete search space.

$$\min(f) \quad s.t. \quad (23)$$

$$\Omega \leq UB \quad (24)$$

$$\Omega \geq LB \quad (25)$$

$$\Omega \in S \quad (26)$$

UKF Design and Optimization

The design of the *UKF* is carried out following the V-model process depicted schematically in Fig. 1. As the vehicle operates along a wide range of conditions (e.g. motorway, aggressive cornering), the uncertainties associated with the plant and measured signals may change considerably depending on the driving situation Kiencke and Nielsen (2005). Thus, a unique tuning strategy could compromise the performance of the *UKF* in certain circumstances Tuononen (2009). A segmentation approach is proposed in order to isolate different driving patterns and determine the best set of *UKF* parameters using metaheuristic optimization. In this paper, this process is particularized for *steady-state* and *transient* driving maneuvers. Nevertheless, the same methodology can be adapted and extended to handle a wider range of driving situations. The subsequent integration of each optimum tuning into an adaptive *UKF* structure and its experimental validation will be pursued in the next research steps.

The flow of the optimization performed to tune the filter at each driving condition is depicted schematically in Fig. 2. First of all, a set of selected maneuvers representative of a driving situation (e.g. steady-state) is generated in the vehicle dynamics simulation software *IPG-CarMaker*®. These runs are concatenated to form the tuning dataset of the *UKF*. The offline optimization of the *UKF* is carried out in the following manner. At each iteration step, the *UKF* is initialized with the set of *UKF* parameters (Ω , see Section “Objective Function Formulation”) obtained in the preceding iteration. The *UKF* is then simulated using the filter inputs (U) and filter

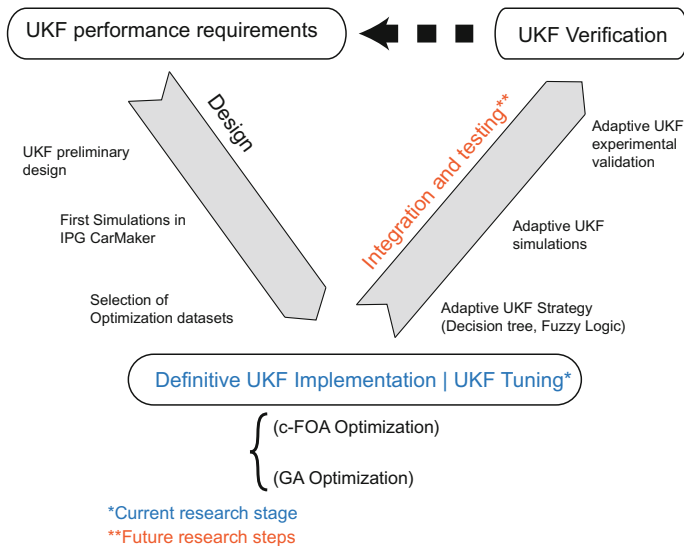


Fig. 1 V-Model for UKF design

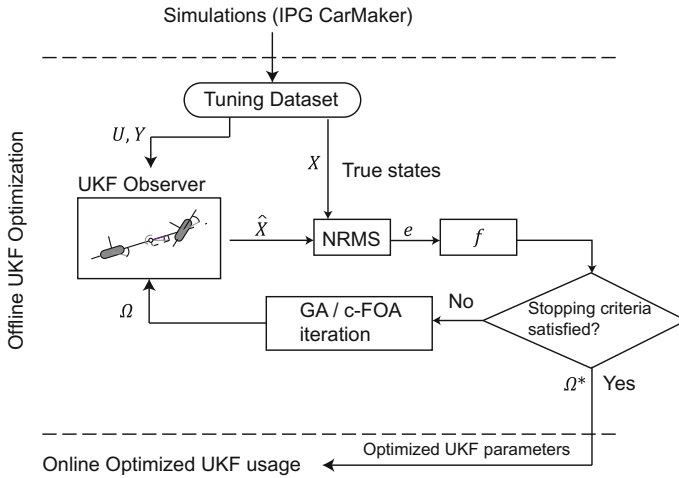


Fig. 2 Optimization flow diagram

measurements (Y) taken from the tuning dataset, and the $NRMS$ errors (e) of the states estimated by the filter (\hat{X}) are calculated. The objective function (f) is evaluated using these errors and, if the stopping criteria is fulfilled, the optimization is finished. The vector of optimized parameters (Ω^*) is obtained and the UKF is run online employing these values. Otherwise, the iterative process is continued until the stopping criteria is met.

Selection of Tuning Datasets

The catalogue of maneuvers presented in Table 1 was proposed to optimize and validate the UKF . These maneuvers are common standardized tests that are often executed in proving grounds to characterize the chassis behavior. The training datasets were formed by a reduced number of maneuvers representative of each driving situation with the aim to limit the computational time required to perform the optimization, and the rest of runs were used at the validation stage. The selection of the maneuvers conforming the dataset was carried out based on the experience of the authors in automotive testing and chassis characterization. Specifically, the maneuvers marked with an X in the *Tuning* column of the Table 1 were used to optimize the UKF (tests #1, #2 in the steady-state dataset, #6, #8, #12 in the transient dataset), while the ones marked with an X in the *Val.* (validation) column were used to validate the UKF . These maneuvers were generated in *IPG-CarMaker*® employing a C-segment-like vehicle and a MF 6.1 tire model (see Acosta and Kanarachos (2016) for further details). An additive white-gaussian noise model was employed to reproduce the uncertainties associated with the measurement equipment RaceLogic (2015). The

Table 1 Catalogue of Maneuvers for UKF Performance evaluation. *SS: Steady-state, TR*: Transient, SSR*: Constant Radius, DS*: Dwell sine, LC*: Lane Change, MS*: Maintain speed, CD*: Coast down, PB*: Partial braking, HB*: Hard Braking

Test	Group	Tuning	Val.	Test	Group	Tuning	Val.
#1 SSR-40m, Gas 80%	SS	X		#7 DS-90deg CD	TR		X
#2 SSR-100m, Gas 50%	SS	X		#8 DS-90deg PB	TR	X	
#3 SSR-100m, Gas 80%	SS		X	#9 DS-90deg HB	TR		X
#4 SSR-80m, Gas 80%	SS		X	#10 ISO-LC MS	TR		X
#5 SSR-60m, Gas 50%	SS		X	#11 ADAC-LC CD	TR		X
#6 DS-150deg CD	TR	X		#12 SLALOM-36m	TR	X	
-	-	-	-	#13 SLALOM-18m	TR		X

UKF was implemented in *Simulink*® and the discretization time was set to 1ms. The measured signals were acquired using a sampling frequency of 100Hz and a zero-order hold approach.

UKF Optimization

The *UKF* was optimized and compared to a *base* configuration obtained following an iterative trial and error process. Two optimizations were performed, one employing the *steady-state* tuning dataset (f_{SS}) and one using the *transient* tuning dataset (f_{TR}). The results obtained after completing the optimizations are presented in Table 2. The performance of the tuned *UKFs* under maneuvers not included in the tuning dataset is studied in the next section.

The *c-FOA* routine Kanarachos et al. (2017) was implemented in Matlab and the automated tuning sequence depicted in Fig. 2 was followed. Several simulations were carried out varying the flies' population size in order to study the convergence of the algorithm.

Concerning Genetic Algorithms (*GA*), the *ga* routine from the *Genetic Algorithm Toolbox* (Matlab ®) was employed to optimize the *UKF*. A structured *GA* optimization approach was proposed in this paper. During the initial stage of the optimization, solutions with a poor resolution are pursued. The search space is discretized and an integer-based optimization problem is solved to locate the domain region where the best values are likely to be. Once a potential domain region has been identified, the *UKF* tuning is formulated as a real-valued constrained *GA* optimization problem, where upper and lower bounds are imposed around the sub-optimal values found

Table 2 Numerical values of the objective functions using different optimization routines

Proposed algorithms	f_{SS}	f_{TR}	Other algorithms	f_{SS}	f_{TR}
Base (Trial and Error)	0.250853	0.052363	SQP	0.103432	0.047492
GA integer	0.102446	0.045409	Nelder-Mead	0.244358	0.044852
GA real-valued	0.094092	0.044696	ABC	0.226566	0.231754
c-FOA	0.116444	0.045630	PSO	0.212778	0.049918
			DE	0.549621	0.222337

in the integer-based optimization. Concerning the integer-based stage, the discrete sets $S_1 = \{100, 10, 1, 0.1 \dots, 1e - 5\}$ and $S_2 = \{1, 1e - 2, 1e - 4\}$ were selected for the design variables (Q , R), and (α) respectively. The rest of *UKF* parameters were set to their recommended values Wan and Van Der Merwe (2000). The cross-over fraction was set to 0.8 and the scale and shrink mutation coefficients were left as default to limit the degrees of freedom of the optimization. The same process was repeated during the real-valued constrained optimization. Overall, the convergence of the best individuals was found after a reduced number of generations. As can be noticed in Table 2, the improvement achieved during the second optimization stage (real-valued) is very small, being the integer-based optimization enough to achieve an acceptable solution.

The optimization was repeated employing other optimization routines with the aim to benchmark the performance of *c-FOA* and *GA*. Specifically, Sequential Quadratic Programming (*SQP*), *Nelder-Mead*, Artificial Bee Colony (*ABC*), Particle Swarm Optimization (*PSO*), and Differential Evolution (*DE*) algorithms were compared to *GA* and *c-FOA*. As can be noticed, only *SQP* was able to match the performance exhibited by the proposed algorithms. Concerning *Nelder-Mead* and *PSO*, reduced errors were obtained when the transient-dataset was employed, but the algorithms failed to reduce the error of the steady-state dataset. Finally, the worst performance was exhibited by the *ABC* and *DE* optimization algorithms.

Due to space limitations, it is not possible to describe in detail the algorithms. The results were achieved for the same number of function evaluations. The difference between *SQP* and the rest algorithms was that *SQP* was initiated using the manual tuning result, while the population-based ones, randomly. From a computational point of view, population-based algorithms have the potential to run faster when processed in parallel and thus achieve the best value faster. Furthermore, the results indicate that possibly a smaller population size but larger number of iterations would increase the performance of population-based algorithms.

Results

The relative improvement (27) of the optimized *UKFs* (*SS* for steady-state tuning and *TR* for transient tuning) with respect to the base setup is depicted in Fig. 3.

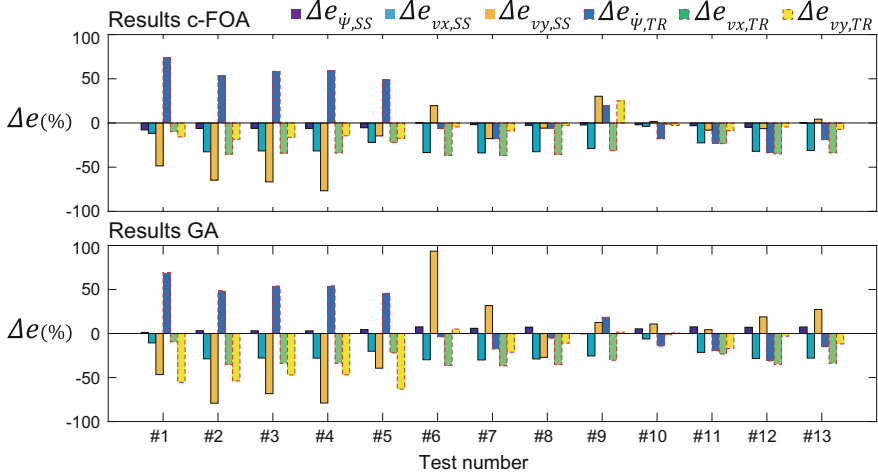


Fig. 3 Relative improvement of the optimized *UKFs* with respect to the base configuration

The results have been computed for all the maneuvers and configurations optimized in order to check whether a single configuration (*SS* or *TR*) is suitable for all the driving situations and a constant covariance *UKF* configuration can be implemented.

$$\Delta e = 100 \frac{e_{opt} - e_{base}}{e_{base}} \tag{27}$$

Overall, the error reduction achieved with the optimized filters is remarkable, particularly in the lateral velocity error of the *UKF* optimized in *SS* conditions (up to 60% reduction on the *NRMS* error) during the validation in steady-state maneuvers (#1 – #5). Concerning the transient maneuvers (#6 – #13), a consistent reduction in the yaw rate and longitudinal velocity errors is noticeable with the transient-optimized *UKFs*. Such improvements indicate that apart from the filter structure selected during the filter design stage (Fig. 1), the filter optimization is of crucial importance (which is often disregarded in the literature). Finally, results suggest that a trade-off solution would be difficult to obtain, as the yaw rate error increases considerably when the transient-optimized *UKFs* are tested in *SS* conditions and the lateral velocity error of the steady-optimized *UKF* augments in *TR* tests, particularly for the *GA*-optimized *UKF*. Fig. 4 depicts the results obtained during the validation of the optimized *UKF* in steady-state maneuvers (test #4, Steady-state 80-meter constant radius maneuver). As can be noticed, an important offset is observed in the lateral velocity with the base configuration. In these circumstances (quasi-static driving conditions) the difference between the front and rear axle lateral forces is minimum (yaw acceleration ≈ 0), and the lateral velocity estimation is very sensitive to the uncertainties associated to the tire lateral forces (e.g. camber gain due to the steering axis inclination). The optimized *UKFs* outperform the base configuration, reducing drastically the lateral

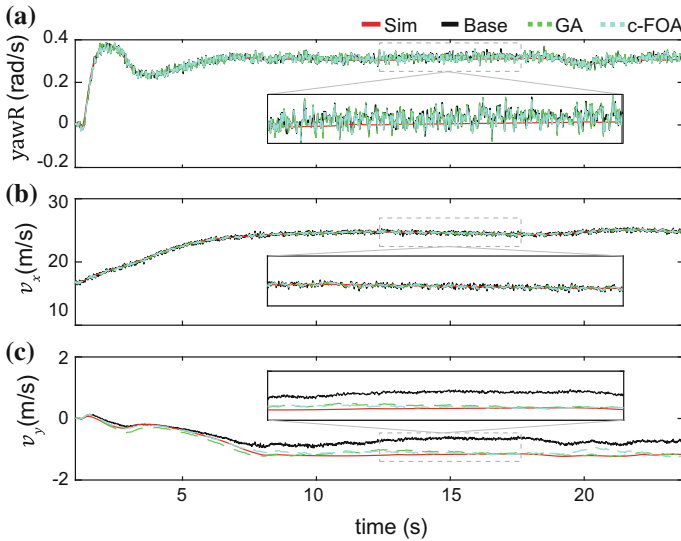


Fig. 4 Test #4, **a** yaw rate, **b** longitudinal velocity, and **c** lateral velocity

velocity error (Fig. 4c). The results obtained during the evaluation of the optimized *UKFs* in transient situations (test #13, 18-meter slalom test) are depicted in Fig. 5. In this case the improvement is less noticeable because the base configuration already provides acceptable results. Nevertheless, a considerable reduction in the noise level of the longitudinal velocity can be appreciated in Fig. 5b.

Metrics

Finally, the *NRMS* errors of the optimized *UKFs* are presented in Tables 3–4. The errors of the *base* tuning have been added for comparative purposes. In what concerns the steady-state errors, these keep below the 10% band for the majority of the tests performed. Values above this band are observed in tests (#1, #5), but these can be accepted taking into account that the initial errors seen with the base configuration are significantly higher (≈ 40 and 23% respectively). Yaw rate and longitudinal velocity errors are slightly lower for the *c-FOA* optimization while the *GA* optimization provides better lateral velocity estimates. Regarding the results obtained in the transient tests, these are below the 10% for all the tests. In this case, negligible differences are identified between *c-FOA* and *GA* optimizations. In addition, the improvement seen with the optimized *UKFs* is less noticeable because acceptable results were obtained with the base configuration. Nevertheless, a considerable time and effort were required to obtain manually the filter parameters of the latter configuration, while the optimized *UKF* values were generated automatically.

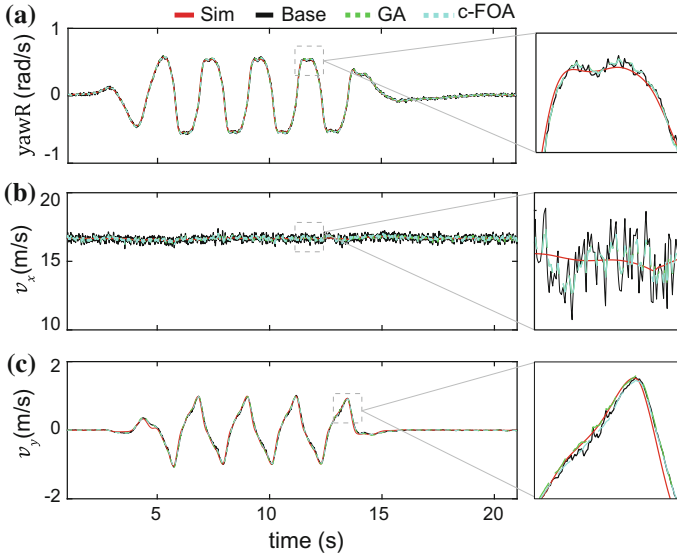


Fig. 5 Test #13, **a** yaw rate, **b** longitudinal velocity, and **c** lateral velocity

Table 3 NRMS errors (%) in steady-state maneuvers, SS optimized UKFs and base tuning

Test	Base			GA			c-FOA		
	$e_{\dot{\psi}}$	e_{v_x}	e_{v_y}	$e_{\dot{\psi}}$	e_{v_x}	e_{v_y}	$e_{\dot{\psi}}$	e_{v_x}	e_{v_y}
#1	3.24	0.92	40.11	3.28	0.83	21.40	2.98	0.82	20.64
#2	4.38	0.59	23.43	4.52	0.43	4.87	4.11	0.40	8.30
#3	4.56	0.60	28.32	4.70	0.44	8.97	4.29	0.42	9.45
#4	3.70	0.67	30.06	3.82	0.49	6.33	3.47	0.46	7.01
#5	3.44	0.79	23.22	3.60	0.63	14.07	3.26	0.62	19.82

Table 4 NRMS errors (%) in transient maneuvers, Transient optimized UKFs and base tuning

Test	Base			GA			c-FOA		
	$e_{\dot{\psi}}$	e_{v_x}	e_{v_y}	$e_{\dot{\psi}}$	e_{v_x}	e_{v_y}	$e_{\dot{\psi}}$	e_{v_x}	e_{v_y}
#6	1.79	0.84	2.49	1.72	0.53	2.62	1.68	0.53	2.38
#7	2.16	0.84	2.24	1.78	0.53	1.76	1.77	0.53	2.03
#8	2.31	0.85	4.69	2.18	0.55	4.17	2.17	0.55	4.55
#9	1.10	0.93	5.91	1.31	0.65	6.00	1.31	0.64	7.38
#10	2.65	0.66	2.79	2.27	0.66	2.81	2.17	0.66	2.72
#11	2.53	0.67	2.37	2.04	0.51	1.97	1.96	0.52	2.16
#12	4.37	0.76	6.60	3.01	0.50	6.40	2.90	0.49	6.30
#13	2.83	0.85	4.45	2.41	0.56	3.93	2.30	0.56	4.13

Conclusions

In this paper, an automated procedure to optimize an Unscented Kalman Filter for vehicle dynamics state estimation has been proposed using a Genetic Algorithm and Fruit Fly optimization. Results demonstrate the suitability of these optimization approaches to enhance the performance of the Kalman Filter, particularly during steady-state maneuvers. In addition, the tuning procedure has been achieved using a short training dataset composed of a reduced number of driving maneuvers, thus contributing to the overall reduction of the optimization time. The improvement achieved with the optimized UKFs remarks the importance of the Kalman Filter tuning, which has been often disregarded in the literature and performed manually in an ad hoc fashion to achieve “good-enough” results. The proposed approach not only reduced significantly the tuning time but also provided quasi-optimized solutions. Moreover, authors envisage that the adoption of parallel processing tools in the proposed optimization will increase notably the productivity of the tuning process. Apart from that, results indicate that a trade-off tuning configuration may be unsuitable for an accurate estimation, and a covariance-scheduling strategy could be beneficial to reduce the estimation error during changing driving conditions.

During the following stages of this research, the complete integration of the *UKF* in an adaptive manner and its experimental validation with real measurements (eliminating the white Gaussian noise assumption) will be pursued.

Acknowledgements This research is part of the Interdisciplinary Training Network in Multi-actuated Ground Vehicles (ITEAM) and has received funding from the European Unions Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No: 675999.

References

- Acosta M, Alatorre A, Kanarachos S et al (2017) Estimation of tire forces, road grade, and road bank angle using tire model-less approaches and Fuzzy Logic. World congress of the international federation of automatic control
- Acosta M, Kanarachos S (2016) Tire force estimation and road grip recognition using extended kalman filter, neural networks, and recursive least squares. *Neural Computing and Appl*, Springer 2017:1–21
- Åkesson B M, Jørgensen J B, Poulsen N K et al (2007) A Tool for kalman filtering tuning. 17th European symposium on computed aided engineering
- Antonov S, Fehn A, Kugi A (2011) Unscented Kalman filter for vehicle state estimation. *Vehicle Syst Dyn* 49(9):1497–1520
- Chipperfield A, Fleming P, Pohlheim H et al (2009) Genetic algorithm toolbox, for use with Matlab. In: Department of automatic control and systems engineering University of Sheffield
- Doumiati M, Charara A, Victorino A, et al (2012) Vehicle dynamics estimation using kalman filtering, Wiley-ISTE
- Doumiati M, Victorino A, Charara A (2009) Estimation of vehicle lateral tire-road forces: a comparison between extended and unscented Kalman filtering. European control conferences

- Eom J, Lee H, Choi B (2014) A study on the tire deformation sensor for intelligent tires. *Int J Precis Eng* 15(1):155–160
- Gadola M, Chindamo D, Romano M et al (2014) Development and validation of a kalman filter-based model for vehicle slip angle estimation. *Vehicle Syst Dyn* 52(1):68–84
- Goodall C, El-Sheimy N (2007) Intelligent tuning of a kalman filter using low-cost MEMS inertial sensors. In: *International symposium on mobile mapping technology*
- Haudum M, Hohensinn F, Höll M, et al (2016) Vehicle State Estimation from a Sports-car application point of view focusing on handling dynamics. In: *International symposium on advanced vehicle control*
- Haupt R.L., Haupt S.H. (2004) *Practical genetic algorithms*, 2nd ed, Wiley
- Hrgetic M, Deur J, Ivanovic V et al (2014) Vehicle sideslip angle EKF estimator based on nonlinear vehicle dynamics model and stochastic tire forces modeling. *SAE Int J of Passeng Cars - Mechanical Syst*
- Kanarachos S (2013) Design of an intelligent feed forward controller system for vehicle obstacle avoidance using neural networks. *Int J Vehicle Syst Modell Test* 8(1):55–87
- Kanarachos S (2014) A new min-max methodology for computing optimized obstacle avoidance steering maneuvers of ground vehicles. *Int J Syst Sci* 45(5):1042–1057
- Kanarachos S, Griffin J, Fitzpatrick E (2017) Efficient truss optimization using the contrast-based fruit fly optimization algorithm. *Comput Struct* 182:137–148
- Kaur N, Kaur A (2016) Tuning of extended kalman filter for nonlinear state estimation. *IOSR J Computer Eng* 18(5):14–19
- Kaur N, Kaur A (2016) A review on tuning of extended kalman filter using optimization techniques for state estimation. *Int J Comput Appl* 145(15):1–5
- Kiencke U, Nielsen L (2005) *Automotive control systems: for engine, driveline, and vehicle*, Springer
- Klier W, Reim A, Stapel D (2008) Robust estimation of vehicle sideslip angle—an approach w/o vehicle and tire models. In: *Proceedings of the SAE world congress*
- Koch P.A. (2011) *Adaptive control of mechatronic vehicle suspension systems*, PhD thesis. Technische Universität München
- Matsuzaki R, Keating T, Todoroki A (2008) Rubber-based strain sensor fabricated using photolithography for intelligent tires. *Sens Actuators A: Phys* 148(1):1–9
- Pacejka HB (2012) *Tire and vehicle dynamics*, butterworth-heinemann
- Pan W (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. *Knowl-based Syst* 26:69–74
- RaceLogic (2015) RLVBIMU04 inertial motion unit technical datasheet, RaceLogic
- Rhudy M, Gu Y (2013) Understanding nonlinear kalman filters, part II: an implementation guide. *Interact Rob Lett Tutorial*
- Scardua L, da Cruz J (2016) Automatic tuning of the unscented kalman filter and the blind tricyclist problem: an optimization problem. *IEEE Control Syst* 36(3):70–85
- Ting TO, Man KL, Lim EG et al (2014) Tuning of kalman filter parameters via genetic algorithm for state-of-charge estimation in battery management system. *Scientific World J, Hindawi*
- Tuonen A (2009) Vehicle lateral state estimation based on measured tire forces. *Sens* 9(11):8761–8775
- Wan EA, Van Der Merwe R (2000) The unscented kalman filter for nonlinear estimation. In: *Adaptive symposium for signal process, communications, and control symposium*

Optimization of Ascent Assembly Design Based on a Combinatorial Problem Representation



Michael Hellwig, Doris Entner, Thorsten Prante, Alexandru-Ciprian Zăvoianu, Martin Schwarz and Klara Fink

Abstract The paper addresses the integration of optimization in the automated design process of ascent assemblies. The goal is to automatically search for an optimal path connecting user defined inspection points while avoiding obstacles. As a first step towards full automation of the ascent assembly design, a discrete 2D model abstraction is considered. This establishes a combinatorial optimization problem, which is tackled by the use of two distinct strategies: a greedy heuristic and a genetic algorithm variant. Applying modeling approach and algorithms to multiple test cases, partly artificial and partly based on a manufactured crane, shows that the automated ascent assembly design tasks can successfully be enhanced with optimal path planning.

M. Hellwig (✉)

Research Centre Product and Process Engineering, Vorarlberg University of Applied Sciences, Hochschulstr. 1, 6850 Dornbirn, Austria
e-mail: Michael.Hellwig@fhv.at

D. Entner · T. Prante

Design Automation, V-Research GmbH – Industrial Research and Development, Stadtstr. 33, 6850 Dornbirn, Austria
e-mail: Doris.Entner@v-research.at

T. Prante

e-mail: Thorsten.Prante@v-research.at

A.-C. Zăvoianu

Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria
e-mail: Ciprian.Zavoianu@jku.at

M. Schwarz · K. Fink

Technology Management, Liebherr-Werk Nenzing GmbH, Dr.-Hans-Liebherr-Str. 1, 6710 Nenzing, Austria
e-mail: Martin.Schwarz@liebherr.com

K. Fink

e-mail: Klara.Fink@liebherr.com

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_19

Introduction

The automation of engineering design processes offers great potential for improving performance by automating tasks that can be standardized and that are frequently repeated (Verhagen et al. 2012). To this end, the design process can be simplified and accelerated or existing designs can be enhanced. Integrating optimization procedures into the automated design process is supportive in both directions, e.g. by reducing design time, time-to-market and construction costs as well as leading to improved and potentially unexpected designs and thus, increasing the diversity of valid designs.

In this paper, we consider the automatic and optimized design of ascent assemblies. The composition of such ascent assemblies typically involves a set of standardized, parameterizable components (e.g. platforms and ladders of varying dimension). The only prerequisite to ensure an optimization potential for the walkway design is a sufficient number of degrees of freedom for positioning the path.

There is a wide range of application areas, including ascent assemblies for building facades, warehouses, and inspection purposes on machines or vehicles. We here focus on the industrial use case of ascent assemblies for cranes, two examples of which are shown in Fig. 1.

In a previous project, in cooperation with Liebherr-Werk Nenzing (LWN 2017), the design process for ascent assemblies of cranes was automated by developing a software called *Automated Crane Component Design* (ACC-Design) (Frank et al. 2014). A part of this software allows the designer to create various forms of platforms, ladders, and stairs by specifying a reduced number of necessary input parameters. Supported by ACC-Design, the engineer can then combine these components according to the respective definitions. The complete ascent assembly design is translated into a 3D-CAD model, which can be attached to the original crane structure. Furthermore, ACC-Design generates production drawings, bills of materials and costs of the assembly.

The planning of a suitable path for the ascent assembly is left to the designer. There is usually a considerable potential for optimization in the manual design, as

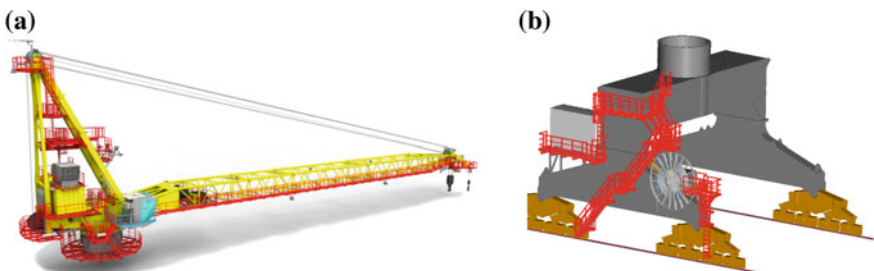


Fig. 1 Demonstration of ascent assembly inspection walkways (highlighted in red) for cranes: in **a** an offshore crane is shown, in **b** a gantry of a mobile harbour crane

it is simply impossible to take into consideration all possible combinations. Further, manual planning reduces the likelihood that innovative design ideas will emerge.

In this paper, an optimization procedure is suggested to automate the planning of the ascent assembly composition. Thereby, an important step towards full automation of the ascent assembly design is taken. For this purpose, the problem has to be formalized in a way that a solver independently determines the (near) optimal connection of previously specified access points (e.g. for maintenance) while avoiding potential obstacles. The design task is thus reduced to the definition of

- (I) the required access points on the crane surface, and
- (II) the excluded points (obstacles) on the crane surface.

Involving optimization into the design automation process requires a suitable problem formalization and the specification of an appropriate optimization goal (e.g. minimize distance or cost). The way of modeling directly affects the choice of solver and consequently the solution quality and variety. This paper provides one suggestion for modeling the problem, in particular, by discretizing the design space. Based on this representation, a suitable combinatorial optimization problem is introduced. The optimization is then carried out using two distinct approaches: a greedy heuristic and a genetic algorithm variant. The algorithms are applied to six test cases of varying complexity, three of which are artificial ones and three of which are based on the gantry of the crane in Fig. 1b. The paper concludes with a discussion of the observations and gives an outlook on future research directions.

Problem Modeling and Formalization

To formalize the optimization problem, we need to define an objective function and a suitable search space representation. As an initial goal of the optimization, we confine ourselves to the subject of minimizing the ascent assembly length. At least for the considered crane models, simulations suggest that the walkway length can be directly linked to the construction costs. However, costs of combining different types of ascent assemblies are neglected.

Considering the search space representation of the optimization problem, different approaches of modeling will have their pros and cons. In this regard, a continuous formulation requires the identification of multiple constraints (with respect to feasible inclination angles, minimal and maximal component dimensions, exclusion of consideration of obstacle areas, etc.). Moreover, post-processing of the continuous optimization result by discretization might be necessary to ensure that the computed design complies with ergonomic norms. On the other hand, a combinatorial problem formulation comes with the need to introduce an appropriate discretization of the crane surface.

In this paper, an abstraction to represent the ascent assembly design problem as a combinatorial optimization problem is conducted. To this end, the problem complexity is reduced by making the following assumptions: Only the lateral surfaces

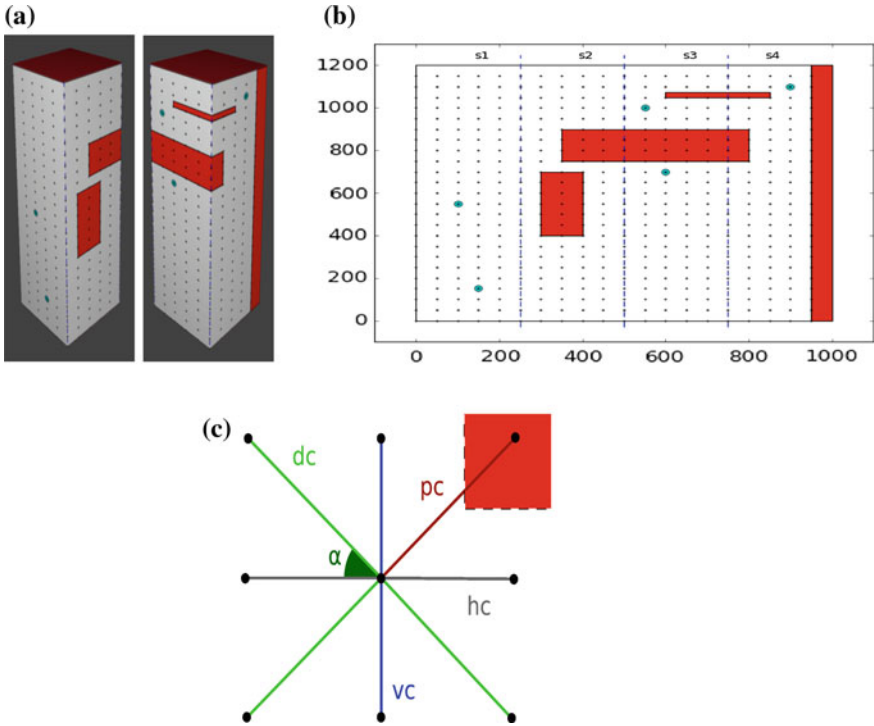


Fig. 2 **a** 3D representation of a simple cuboid surface; **b** unfolded lateral cuboid surface in 2D - the cuboid surface is separated into four segments $s1$ to $s4$ representing the respective sides. The green dots display user specified access points while the red rectangles illustrate obstacle areas. **c** The eight connecting edges of a central node to its adjacent nodes

of the abstract crane model, cf. Fig. 2a, are considered. As shown in Fig. 2b, the surface can be unrolled to a two-dimensional rectangle. The optimization is then performed within a two-dimensional search space defined by $(x, y) \in (0, 1000) \times (0, 1200) \subset \mathbb{R}^2$. The goal is to determine the optimal path between several user-defined access points on this surface while avoiding previously specified obstacles.

We limit the path composition to three basic types of components, whose quantities, dimensions and ordering have to be determined. These component types are: rectangular platforms (horizontal connections), ladders (vertical connections), and stairs (diagonal connections). Our component selection is based on the standardization of the ascent assembly design components used by the ACC-Design software. It ensures that our formalization approach is practicable in real-world applications. Furthermore, each component type has minimum and maximum dimensions, individual costs, and other specifications such as angle of inclination, railings, bails or various levels of solidness. Regarding the latter options, we are initially limiting ourselves to common standard components.

Taking into account the obstacle areas usually transforms the problem into a constrained, non-linear optimization problem regardless of the objective function. This naturally increases the problem complexity. One way to avoid constraints and to maintain a linear problem structure is modeling the scenario as a graph-based combinatorial optimization problem. For this purpose, a grid is generated on the two-dimensional surface. Such a grid is displayed in Fig. 2a, b by use of the black dots. Each grid point resembles an optional attachment point on the crane structure. For a realistic representation of the test case, the grid structure should ensure the following requirements:

- The distances of adjacent points should correspond to admissible lengths of the design parts (e.g. the length of a platform). In general, the smaller the point distances, the more combinatorial options exist in the search space. On the other hand, very short connections may object to ergonomic requirements (e.g. ladders of size less than one meter are not convenient for human beings).
- The maximum point distances should not exceed industrial norms (e.g. transportation length).
- The diagonal point distances should be determined according to feasible inclination angles for the staircases. It can be ensured by calculating the vertical distances with respect to predefined horizontal distances and a suitable angle of inclination.
- Finally, the user-specified access or inspection points must be located on the grid. If the given coordinates do not correspond to a specific grid point (unlike Fig. 2), the software should automatically specify a grid which allows a person to reach the desired locations without additional effort.

The grid points can be regarded as the set of nodes (or vertices) V of a mathematical graph $G(V, E, c)$. The corresponding set of edges E is then determined by the connection possibilities between these nodes. In order to limit the number of edges $|E|$, we only consider edges of adjacent vertices. The weights c of the edges are linked to the Euclidean distance of the respective connection. For a detailed explanation of graphs refer to Bondy et al. (2008).

The connection possibilities of a central attachment point of the area are shown in Fig. 2c. A central node of the graph $G(V, E, c)$ has eight edges. The four diagonal edges are equipped with the weights dc , referring to the length of a single diagonal component of the ascent assembly structure. The two horizontal edges get the weights hc , and the two vertical edges are equipped with weights vc , respectively. As the rectangular area in Fig. 2b represents the lateral surface of a folded-up cuboid, the nodes of the lowest level have no downward connections and consequently only 5 edges. The same applies along the upper boundary. The horizontal boundaries are permeable, i.e. direct edges from nodes with x-coordinate 1000 (right) to nodes with x-coordinate 0 (left) exist. In order to include the obstacles (red areas) into the modeling of the graph, edges crossing the obstacle areas are accounted with comparably large penalty costs pc , see Fig. 2c. This way, paths across the obstacles are avoided within the optimization.

The prior deliberations result in the formulation of a combinatorial optimization problem. To this end, the set of all required access points (also referred to as terminal

nodes) is denoted by S . By interpreting the grid on the lateral surface of the crane as a graph $G(V, E, c)$, the objective turns into finding a connected subgraph $G'(V', E', c)$ of $G(V, E, c)$ that contains all access points of $S \subset V' \subset V$, and that minimizes the sum of its edge weights $\sum_{e \in E'} c(e)$.

This optimization problem is known as the *Steiner Tree Problem* (STP) in graphs (Gilbert and Pollak 1968). A Steiner tree is a tree in $G(V, E, c)$ that spans $S \subset V$. Considering the STP, a large number of publications exist that address this kind of optimization problem. Strategy recommendations vary from greedy-like algorithms to the use of Evolutionary Algorithms (EAs). Among others, especially in the context of EAs, the following publications have to be noted (Bezenšek and Robič 2014; Huy and Nghia 2008; Kapsalis et al. 1993). As a combination of the (*non-negative*) *shortest path* (SP) and the *minimal spanning tree* (MST) problem, the STP in graphs represents an NP-hard optimization problem, i.e. it cannot generally be solved efficiently in polynomial time. Consequently, for high dimensional problems the optimization goal is usually the computation of a sufficiently good and feasible solution in reasonable time. The present work considers two distinct approaches which are presented in detail within the next section.

Optimization Approaches

In this section, the focus is on two possible approaches to tackle the Steiner Tree Problem with respect to our formalization of the ascent assembly design problem. The first one aims at generating a reasonable solution to the STP by successive shortest path calculation by repeatedly using Dijkstra's algorithm (Bondy and Murty 2008). In this sense, it represents a greedy strategy. The second approach applies a customized Genetic Algorithm (GA) (Sivanandam and Deepa 2008) to a binary representation of the graph. It approaches the optimizer by successively generating a population of subgraphs $G'(V', E', c)$ of $G(V, E, c)$ that represent candidate solutions of the STP. Due to their very intuitive working principles, and due to the success of GAs on comparable Steiner Tree problem instances, e.g. in the context of network planning or microprocessor design (Frommer and Golden 2007), in a first attempt these two methods are considered for proving our formalization concept.

Successive Shortest Path Generation

The first approach for finding a near optimal, feasible path between the user defined inspection points in S is provided by the strategy in Algorithm 2. We refer to this method as *Successive Shortest Path Generation* (SSPG). It successively constructs a subgraph $G'(V', E', c)$ of $G(V, E, c)$ that represents a partial solution of the problem. Step-by-step, the subgraph G' is enlarged until it includes all required nodes of the set S . Hence, the final subgraph $G' \subset G$ represents a feasible solutions. However, the

quality of the obtained solution depends on the ordering of the terminal nodes as well as on the structure of the graph $G(V, E, c)$. That is, it is by no means secured that the generated solution is close to the problem's optimum. Since the SSPG is operating efficiently, this drawback is slightly remedied by executing $\#runs$ repeated runs of the strategy, and re-ordering of the terminal nodes S randomly before each run. The algorithm compares the outcome of two consecutive runs with distinct orderings and stores the best result so far. Optionally, the user may assign an ordering of the inspection points (with respect to their priority) in the beginning.

Let $w_i \in S$, $i \in \{1, \dots, k\}$ represent the i th terminal node in the randomly ordered set S . The SSPG strategy then solves the Shortest Path Problem (SPP) according to the first two access points in the set S . We refer to this operation by $SPP(w_1, w_2)$. That is, the shortest path Y from w_1 to w_2 in $G(V, E, c)$ is calculated by use of Dijkstra's algorithm. The nodes and edges of G covered by the shortest path Y build the first representation of the subgraph $G'(V', E', c) \subset G(V, E, c)$. The strategy then calculates all shortest paths from G' to the next terminal node $w_3 \in S$. The path Y' which has minimal sum of weights adds its nodes to the subgraph $G' \leftarrow G' \cup Y'$. Iterating over the number of inspection points k , Algorithm 1 provides a feasible solution of the STP. After a random permutation of the terminal points, the procedure is repeated. After each repetition, the length of a subgraph G' is compared to the best-so-far solution G_{bsf} . The best current subgraph G_{bsf} is replaced if a shorter path between all terminal nodes of S is identified, i.e. if

$$f' = \sum_{e \in E'} c(e) \leq \sum_{e \in E_{bsf}} c(e) = f_{bsf}, \quad (1)$$

with $E_{bsf} \subset G_{bsf}$ and $E' \subset G'$.

This approach does not guarantee to result in an optimal solution, but it provides a feasible (and reasonably good) approximation with comparably small computational effort. Further speed up may be realized by replacing Dijkstra's algorithm with more advanced path planning algorithms (Klidbary et al. 2017). The final subgraph G'

Algorithm 1 Pseudo code of SSPG.

```

1:  $G_{bsf} \leftarrow \emptyset$ ,  $f_{bsf} \leftarrow \infty$ 
2: for  $j = 1, \dots, \#runs$  do
3:   Sort all  $k$  terminal nodes  $w_i \in S$  in random order.
4:   Compute  $Y = SPP(w_1, w_2)$ .
5:   Define  $G' \subset G$ :  $G' \leftarrow G' \cup Y$ 
6:   for  $i = 3, \dots, k$  do
7:      $Y' \leftarrow SSP(G', w_i) = \min_{v_j \in G'} SPP(v_j, w_i)$ 
8:      $G' \leftarrow G' \cup Y'$ 
9:   end for
10:  if  $f' \leq f_{bsf}$  then
11:     $G_{bsf} \leftarrow G'$ 
12:  end if
13: end for

```

Algorithm 2 Pseudo code of the GA variant.

```

1: Initialize a random population  $P$  of size  $\lambda$ .
2: repeat
3:   Evaluate each individual in  $P$ .
4:   Store the 'best-so-far' solution  $G_{bsf}$ 
5:   Select the mating pool  $MP$  from the population  $P$ 
6:   Add  $G_{bsf}$  to  $MP$ .
7:   Build new population  $P'$  by
8:   repeat
9:     Draw two individuals from  $MP$  without replacement and
10:    apply Crossover and Mutation
11:   until  $|P'| = \lambda$ 
12:   Replace prior population:  $P \leftarrow P'$ 
13: until termination return  $G_{bsf}$ 

```

represents a good and intuitive candidate solution of the ascent assembly design problem. The realized paths can be regarded as design suggestions, as comparison benchmarks or even as input for the second approach.

Customized Genetic Algorithm Variant

Evolutionary Algorithms (De Jong 2006) are generic population-based heuristic optimization algorithms that are inspired by biological evolution. Among EAs, GAs represent a very popular type, especially in the context of discrete or combinatorial optimization problems. Candidate solutions of the underlying problem usually use an encoding (binary, integer or real values) that reflects certain characteristics about the problem being solved. Making use of operators such as recombination (crossover) and mutation, the initial candidate solution is iteratively evolved in the direction of the optimizer.

This section introduces a GA that tackles the STP mentioned in Section “[Problem Modeling and Formalization](#)” representing a formalization of the ascent assembly design problem. The algorithm is inspired by the GA presented in the work of Kapsalis et al. (1993), but it has been partly modified in order to improve its performance in the considered case. The algorithm encodes potential candidate solutions (i.e. subgraphs of $G(V, E, c)$ that represent not necessarily feasible ascent assembly paths) by use of binary bit strings. Its pseudo code is displayed in Algorithm 2.

The GA evolves an initial set (or population) P of candidate solutions towards feasible and ultimately towards (near) optimal pathways. The population size is denoted by the parameter λ . Referring to Kapsalis et al. (1993), we use $\lambda = 10$ in our applications. Only superior (with respect to short path length realizations) candidate solutions inherit their information to the consecutive generation. To this end, the fitness of the whole population has to be evaluated and the best solution found so far is stored (G_{bsf}). A selection operator determines the best individuals that may

contribute to the mating pool MP . If G_{bsf} is not part of the current population, it is added to the mating pool by default. By randomly drawing a pair of two individuals from the mating pool without replacement, and by applying the variation operators, a new population P' is generated. Variation is performed by uniform crossover and mutation. The respective steps of the GA are explained in detail in the textbfs below. The algorithm terminates after a predefined maximal number of iterations (generations) is exceeded or if no further progress is observed for a certain number of generations. The best-so-far solution is returned and has to be decoded into the graph representation.

Encoding. The GA operates on a chromosomal representation of candidate solutions, i.e. subgraphs $G'(V', E', c)$ of $G(V, E, c)$. It is obtained by identifying each vertex of G with a single component in a binary vector of length $|V|$. Assuming that the number of vertices is n , a binary vector \mathbf{y} of the form

$$\mathbf{y} \in \{0, 1\}^n \quad \text{with} \quad y_i = \begin{cases} 0, & \text{if } v \notin V' \\ 1, & \text{if } v \in V' \end{cases} \quad (2)$$

represents the nodes included in a candidate solution $V' \subset G'$ of the STP. It is required that the terminal nodes S are included in each candidate solution, i.e. $S \subset V'$. This is ensured by explicitly excluding those y_i identified with the vertices in S . Reinserting them after optimization reduces the search space dimension to $m = |V| - |S|$.

Initialization. The GA may be initialized with a completely random population which quality will be constantly improved with the number of generations. Note that speed-up can potentially be reached by seeding one or multiple feasible solutions into the initial population. These population members can contribute to the composition of subsequent candidate solutions. Consequently, the optimizer is approximated rather quickly. Such a seeding can be realized e.g. by calculating MST representations of the graph $G(V, E, C)$, by adding the solution from the SSPG approach of Algorithm 1, or by seeding a manually user specified default walkway.

Evaluation. The selection of candidate solutions \mathbf{y} is based on their quality, also referred to as fitness. The fitness is measured by means of the length of the subgraph G' associated with it. Regarding path minimization, shorter lengths correspond to higher fitness. An individual $\mathbf{y} \in P$ is evaluated as follows: First the subgraph G' induced by the nodes $S \cup V'$, i.e. by the binary representation \mathbf{y} , is generated in G . For each component of the potentially unconnected subgraph G' its Minimal Spanning Tree in G is computed by making use of Kruskal's algorithm (Bondy and Murty 2008). The aggregation of the edge weights of all MSTs determines the length of a candidate solution. For each of the potential $t > 1$ components of an unconnected subgraph G' , a large penalty value linearly dependent on t is added to its length. This way, the GA will immanently reduce the MST to the smallest possible representation. The evaluation step can be summarized as:

$$\text{len}(\mathbf{y}) = \sum_{e \in \text{MST}(G')} c(e) + \text{penalty} \cdot (t - 1). \quad (3)$$

Selection. As selection procedure the so-called Roulette selection is used: Each candidate solution is assigned a selection probability according to its relative fitness with respect to the average fitness of the current population. We then stochastically select candidate solutions (with replacement) to form the mating pool set MP for variation. Hence, solutions of above average quality may contribute multiple times to the mating pool, while below average solutions may not contribute at all.

Variation. A new population of candidate solutions is created by crossover and mutation. The breeding is performed until the offspring population reaches the same size as the parental population. All individuals of the prior population are replaced. The recombination of two individuals from the mating pool is performed by *uniform crossover*. That is, after defining a fixed mixing ratio (e.g. $CR = 0.5$) between the two parents, each bit of an offspring is determined with probability CR by the value of the first parent and with probability $(1 - CR)$ by the value of the second parent.

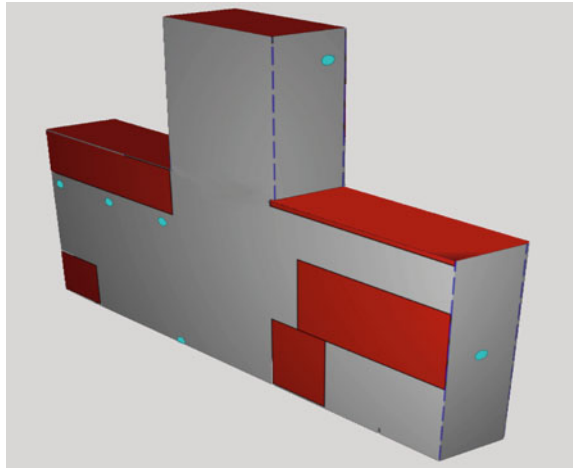
In order to prevent the population from stagnation, random mutations have to be included into the variation step. We try to maintain sufficient genetic population diversity by application of two mutation schemes. The first scheme simply performs *random bit flips* with a probability of $1/m$ for each bit. Here, m is the size of the binary vector \mathbf{y} after exclusion of the required terminal node components. Consequently, in expectation one bit of the binary vector \mathbf{y} is changed. This mutation scheme is applied with probability $\alpha \in [0, 1]$ as long as the population does not include candidate solutions that are identified with connected subgraphs G' (i.e. ascent assembly paths).

At some point, the mating pool will consist of rather similar candidate solutions. In that situation, neither uniform crossover nor random bit flips are able to realize adequate diversity. The solutions will rather be torn into separate components which is most probably degrading their fitness. To this end, another mutation scheme is applied to further maintain diversity after the strategy has evolved the population towards connected solutions. It is especially tailored for the STP in graphs. Instead of randomly flipping single bits in the binary representation of a candidate solutions, it operates in the subgraph representation of \mathbf{y} , i.e. the subgraph \tilde{G} in $G(V, E, c)$ that is induced by \mathbf{y} . First, an individual is destroyed by deleting a path between two randomly selected and connected nodes within the subgraph. The individual is then repaired by replacing the deleted nodes with those of the shortest path between them. Hence, the algorithm is able to contract possible detours with higher probability. If the subgraph \tilde{G} is not connected in the first place, the standard mutation is performed by the random bit flip operator.

Application of the Proposed Methods

In this section, the algorithms introduced in Section “[Optimization Approaches](#)” are applied to six adequately modeled test cases of different complexity. The first three test cases (TC1–TC3) represent variants of the cuboid presented in Fig. 2. Each represents a 2D surface abstraction of a conceptual cuboid, but does not refer to a real-world crane model. Incorporating different numbers of obstacles and varying

Fig. 3 The 3D abstraction of the gantry of a mobile harbour crane, cf. Fig. 1b. The unfolded 2D-representation of the gantry with different entry points is presented in Figs. 4 and 5d, f, and referred to as test case TC4, TC5 and TC6, respectively



inspection point positions, the three test cases require individual grid definitions. As the number of grid points determines the dimension of the search space, this results in different complexities.

The latter three test cases (TC4–TC6) represent an abstraction of the lateral surface of a real-world gantry of a mobile harbour crane of LWN (see Fig. 1b) with three different entry points. To be able to model the gantry with the 2D-representation, two cuboids are stacked on top of each other, cf. Fig. 3. As the three representations only differ by the user defined entry point to the ascent assembly path, they are of the same complexity. However, the inspection point definitions already require a grid of at least 2016 points. By exhibiting only minor specification differences, TC4 to TC6 demonstrate the potential of finding divers design solutions.

The performance results of the SSPG and the GA variant are compared in Table 1. To this end, the table distinguishes the test cases by the minimal number of required grid points (#nodes). Taking into account the number of executed algorithm runs (#runs) the respective results of both strategies are presented in terms of the mean ascent assembly length ‘mean len’ and the length of the best found solution within all runs ‘best len’. The last column displays the time (in seconds) needed to complete all runs. Further, illustrations of the resulting best ascent assembly paths are presented in Figs. 4 and 5.

Considering the SSPG approach (Algorithm 1), Table 1 illustrates its performance on the mentioned test cases. Using Dijkstra’s algorithm, it works very efficiently and is able to provide good solutions with low computational effort. The best solutions on the test cases TC1–TC6 that have been found by SSPG are displayed in Fig. 4. They represent reasonable and intuitive ascent assembly design suggestions. However, taking into account a large number of inspection points k , SSPG cannot guarantee to find an optimal solution in reasonable time. This is due to the increasing number of potential point orderings ($k!/2$).

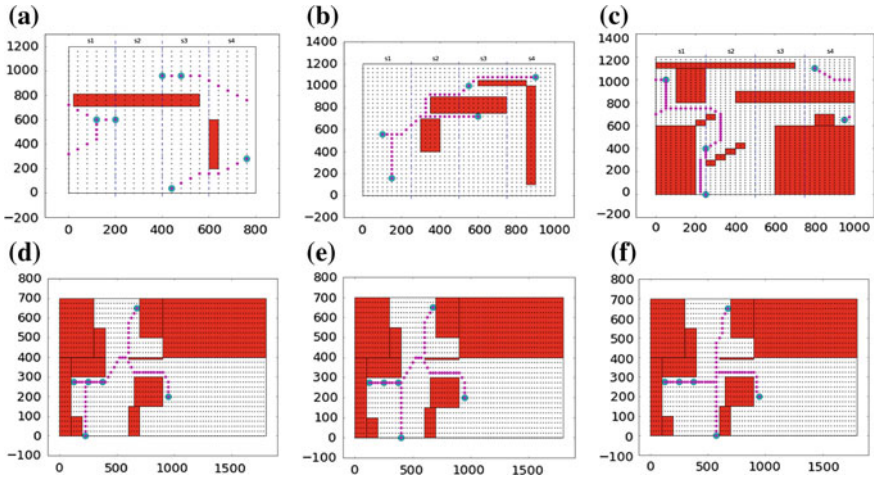


Fig. 4 Illustration of the unfolded lateral surfaces corresponding to the six considered test cases (TC1 – TC6). The magenta point markers display the final ascent assembly path of minimal length that has been found by the SSPG approach

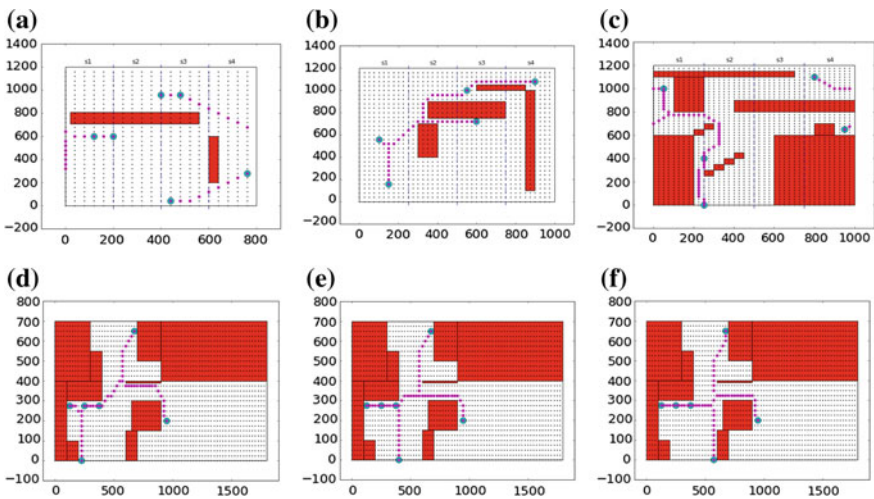


Fig. 5 Illustration of the unfolded lateral surfaces corresponding to the six considered test cases (TC1–TC6). The magenta point markers display the final ascent assembly path of minimal length that has been found by the customized GA variant

Table 1 Results of Algorithms 1 and 2 applied to six test cases presented in Figs. 4 and 5, respectively. For each test case, the table summarizes the search space dimension (“number of nodes, #nodes”), the mean and best fitness results (w.r.t. assembly length) for the specified number of algorithm runs (#runs), as well as the necessary computational time (in seconds) for all #runs

Results of the SSPG approach, Algorithm 1						Results of the GA approach, Algorithm 2					
name	#nodes	#runs	mean len	best len	time	name	#nodes	#runs	mean len	best len	time
TC1	600	100	1665.91	1521.67	72	TC1	600	10	1604.67	1521.67	7577
TC2	1200	100	1954.52	1834.53	130	TC2	1200	10	1875.37	1838.87	12366
TC3	1920	100	1770.20	1730.32	238	TC3	1920	10	1795.99	1736.39	38836
TC4	2016	100	1640.15	1569.97	336	TC4	2016	10	101614.7	1565.69	74465
TC5	2016	100	1634.39	1569.97	337	TC5	2016	10	1678.28	1565.69	74814
TC6	2016	100	1665.12	1612.13	336	TC6	2016	10	1689.16	1603.55	75120

The results in Table 1 also substantiate that the GA variant successfully approaches the optimizer. For the test cases TC1 to TC3, all single runs have been stopped after 50.000 iterations. The iteration budget for the gantry test cases TC4–TC6 has been set to 10^5 iterations. The best ascent assembly paths corresponding to the six test cases are illustrated in Fig. 5. When starting from a completely random population, the use of Algorithm 2 is considerably more time consuming. This does not come as a surprise as the GA needs a large number of iterations for generating a first feasible candidate solution. Additionally, it consumes more time to realize small path changes that become relevant as the algorithm approaches the optimizer. This is due to a decreasing probability of randomly sampling appropriate bits within a large binary vector that still allow for further progress. The second mutation variant is gently counteracting this process and successfully reduces the number of detours in the ascent assembly paths. Instead of starting from a random population, the GA might be initialized with one (or more) feasible path(s). Skipping the effort to create feasible solutions, it is then able to realize improvements on the seeded path rather quickly. Using a random seed path generated by SSPG, the results are briefly presented for TC4 to TC6 in Table 2. In this situation, the GA variant (Algorithm 2) is able to approach the best known solution from Table 1 in almost every run within about 10^4 generations. Hence, a considerable computation time improvement is observable.

Given a large search space dimension (#nodes), the GA variant alone exhibits difficulties to find the optimal solution in reasonable time. However, it is capable to generate comparably good ascent assembly designs from a random initialization, cf. TC1 in Table 1 as well as Figs. 4a and 5a. The respective design observably leaves room for improvements in TC2 and TC3. Deferring the termination, the presented solution would still have improved over time. This is directly substantiated by taking into account the results on the gantry test cases TC4 to TC6. In terms of the necessary number of nodes, these test cases do not differ much from TC3. But having available twice the number of iterations, the GA variant is able to find an improvement over the SSPG results in all three cases. Note that the huge mean value deviation in TC4 is

due to a single run in which no feasible solution was found by the GA, Algorithm 2. All other nine runs resulted in a path length of 1565.69.

Both approaches are capable of finding feasible, (near) optimal ascent assembly paths on all six test cases. Overall, the resulting solutions are on a similar quality level. For the best found solutions further improvements are not immediately obvious. Regarding test case TC1, the GA realizes a slightly different path of similar length.

With respect to computation time, SSPG clearly outperforms the GA variant. This is due to the GA's initially random population and due to the decreasing probability of finding improvements with its variation step, even though the second mutation operator is able to reduce the number of detours. As a result of the immanent shortest path generation, the GA paths are controlled in the direction of the SSPG design. However, the GA's ability to result in new and potentially innovative designs of comparable quality is observable in Fig. 5. On the other hand, given an equal number of algorithm runs the mean path length realized by the GA is potentially closer to the length of the best path. That is, a single run of the GA is expected to result in a better solution than a single run of the SSPG which is heavily relying on selecting the best possible ordering of the inspection points.

Moreover, the GA performance relies on the choice of problem specific strategy parameters that tune the selection and variation behavior. A problem specific empirical study on beneficial strategy parameter settings for the GA remains to be done. On the other hand, there exist more advanced EAs that reduce the computational effort in high dimensional scenarios. For example, parallelization approaches (Huy and Nghia 2008) or making use of adaptive GA variants are likely to result in a speed up. Applying such strategies to our problem formulation is a reasonable direction for future investigations.

Discussion and Way Forward

The present paper focuses on the integration of optimization approaches into the automated ascent assembly design task of cranes. By making use of a grid construction on the (simplified) lateral crane surface, a two-dimensional model abstraction is

Table 2 The results of Algorithm 2 using an initial path generated by SSPG. Corresponding to test cases TC4 – TC6, the table displays the search space dimension (“number of nodes, #nodes”), the initial path length (seed len), the mean and best fitness results (w.r.t. assembly length) for the number of distinct algorithm runs (#runs), as well as the over-all computation time (in seconds) for all #runs

name	#nodes	#runs	seed len	mean len	best len	time
TC4	2016	10	1601.04	1565.69	1565.69	3545
TC5	2016	10	1582.84	1565.69	1565.69	3534
TC6	2016	10	1632.84	1604.16	1603.55	3436

introduced. It is then transferred into a combinatorial optimization problem, in particular the Steiner Tree Problem in graphs. We substantiate the modeling by application of two solvers. The SSPG, Algorithm 1, relying mostly on Dijkstra's algorithm for shortest path generation in graphs, and a customized Genetic Algorithm variant, cf. Algorithm 2. The results from both approaches support the claim that the modeling approach provides reasonable ascent assembly designs.

The used graph representation omits the incorporation of constraints. Despite representing a NP-hard problem, given a beneficial structure, reasonably good solutions of the Steiner Tree problem can be obtained. As the stochastic nature of EAs increases the computational effort, EAs can contribute to higher diversity in the solution set. Providing a variety of good solutions is especially useful from the perspective of design automation and innovative design.

A crucial task for the model abstraction is the definition of the underlying grid structure, which must be performed with respect to industrial norms and corporate definitions. While this involves initial effort, the algorithm output can be directly used as a basis for converting it to CAD models using the existing software ACC-Design. On the other hand, complex structures need a rather high grid granularity which results in large search space dimensions. Consequently, the performance of GA variants is impaired. Future research will have to address the use of more advanced solving strategies. A step in that direction is done in the work of Zăvoianu et al. (2017) where the problem formalization is transferred to a continuous search space. The optimization problem is tackled by a multi-objective evolutionary algorithm revealing good results.

Since the computational effort rises with increasing number of grid points, for future considerations, grid points within obstacle areas should be disregarded during the optimization process. This yields a considerably lower search space dimension and thus runtime improvements. As the results of Table 2 indicate, starting the GA from a randomly determined (most likely infeasible) path representation should be avoided. Instead, a feasible initialization of the path that already connects all required access points is beneficial to gain speed. Considerations along this line, as well as algorithmic adaptations to speed up the GA's performance as mentioned in Section "Application of the Proposed Methods", appear also useful when extending the proposed problem to less regular surfaces or even to 3D crane representations (e.g. needed for modeling the crane in Fig. 1a). Thus, the increasing effort that comes with a search space extension can be mitigated. Once a reasonably specification of access points and connection components has been found, at least the combinatorial problem representation allows for a relatively simple extension to 3D. The optimization approaches preserve their applicability. However, the (automatic) determination of such a graph representation is likely to exhibit its own (yet unidentified) difficulties.

By developing optimization algorithms to automatically determine the path of an ascent assembly, an important step towards *full automation* of the ascent assembly design is taken. As an ultimate goal for future developments, the designer should only need to specify the access points and obstacles in the 3D-CAD model of the crane, from which the abstract representation is generated automatically. This will then be handed over to the optimization algorithm, performing the planning of a

suitable ascent assembly to reach the access points while avoiding the obstacles. The obtained solution shall be represented and, potentially after post-processing by the engineer, be translated into a 3D-CAD model using ACC-Design and finally be attached to the crane model. Thus, the design task for the engineer shall be reduced to specifying access points and obstacles, and verifying the solution.

Acknowledgements The present work is supported by the Austrian Research Promotion Agency FFG through the funding program COMET in the K-Project “Advanced Engineering Design Automation (AEDA)” and by the Austrian Science Fund FWF under grant P29651-N32.

References

- Bezenšek M, Robič B (2014) A survey of parallel and distributed algorithms for the Steiner tree problem. *Int J Parallel Prog* 42(2):287–319
- Bondy J, Murty U (2008) *Graph theory (graduate texts in mathematics)*. Springer, New York
- De Jong KA (2006) *Evolutionary computation: a unified approach*. MIT Press, Cambridge, MA
- Frank G, Entner D, Prante T, Khachatouri V, Schwarz M (2014) Towards a generic framework of engineering design automation for creating complex CAD models. *Int J Adv Syst Meas* 7:179–192
- Frommer I, Golden B (2007) A genetic algorithm for solving the euclidean non-uniform Steiner tree problem. Springer, Boston, pp 31–48
- Gilbert EN, Pollak HO (1968) Steiner minimal trees. *SIAM J Appl Math* 16(1):1–29
- Huy NV, Nghia ND (2008) Solving graphical steiner tree problem using parallel genetic algorithm. In: *IEEE international conference on research, innovation and vision for the future. RIVF 2008*, pp 29–35
- Kapsalis A, Rayward-Smith VJ, Smith GD (1993) Solving the graphical steiner tree problem using genetic algorithms. *J Oper Res Soc* 44(4):397–406
- Klidbary SH, Shouraki SB, Kourabaslou SS (2017) Path planning of modular robots on various terrains using Q-learning versus optimization algorithms. *Intel Serv Robot* 10(2):121–136
- LWN (2017) Liebherr-Werk Nenzing GmbH. <http://www.liebherr.com/en-GB/35267.wfw>. Accessed 03 June 2017
- Sivanandam S, Deepa S (2008) *Genetic algorithms*. Springer, Berlin, pp 15–37
- Verhagen WJ, Bermell-Garcia P, van Dijk RE, Curran R (2012) A critical review of knowledge-based engineering: an identification of research challenges. *Adv Eng Inf* 26(1):5–15
- Zăvoianu AC, Saminger-Platz S, Entner D, Prante T, Hellwig M, Schwarz M, Fink K (2017) On the optimization of 2D path network layouts in engineering designs via evolutionary computation techniques. In: *Proceedings of EUROGEN 2017, ECCOMAS thematic conference, Madrid, Spain*

On the Optimization of 2D Path Network Layouts in Engineering Designs via Evolutionary Computation Techniques



Alexandru-Ciprian Zăvoianu, Susanne Saminger-Platz, Doris Entner, Thorsten Prante, Michael Hellwig, Martin Schwarz and Klara Fink

Abstract We describe an effective optimization strategy that is capable of discovering innovative cost-optimal designs of complete ascent assembly structures. Our approach relies on a continuous 2D model abstraction, an application-inspired multi-objective formulation of the optimal design task and an efficient coevolutionary solver. The obtained results provide empirical support that our novel strategy is able to deliver competitive results for the underlying general optimization challenge: the (obstacle-avoiding) Euclidean Steiner Tree Problem.

A.-C. Zăvoianu (✉) · S. Saminger-Platz

Department of Knowledge-Based Mathematical Systems,
Johannes Kepler University Linz, Altenbergerstraße 69, 4040 Linz, Austria
e-mail: Ciprian.Zavoianu@jku.at

S. Saminger-Platz

e-mail: Susanne.Saminger-Platz@jku.at

D. Entner · T. Prante

Design Automation, V-Research GmbH - Industrial Research and Development,
Stadtstraße 33, 6850 Dornbirn, Austria
e-mail: Doris.Entner@v-research.at

T. Prante

e-mail: Thorsten.Prante@v-research.at

M. Hellwig

Research Centre for Process and Product Engineering,
Vorarlberg University of Applied Sciences, Hochschulstraße 1, 6850 Dornbirn, Austria
e-mail: Michael.Hellwig@fhv.at

M. Schwarz · K. Fink

Technology Management, Liebherr-Werk Nenzing GmbH,
Dr.-Hans-Liebherr-Str. 1, 6710 Nenzing, Austria
e-mail: Martin.Schwarz@liebherr.com

K. Fink

e-mail: Klara.Fink@liebherr.com

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_20

Introduction

In the present work we describe initial results concerning the automatic generation of cost-optimal *complete ascent assembly structures* (CAA-Structures) – external access structures for cranes (as shown in Fig. 1a), building facades, over-sized industrial machines, etc. Fig. 1b shows an example of a CAA-Structure that is itself composed from several types of ascent assembly modules (i.e., sub-assemblies) like rectangular or round platforms, stairs and ladders.

The task of designing individual ascent assembly modules, although important, is rather repetitive and time consuming. In recent years, in light of strong financial and operational incentives, there has been a consistent and successful effort to standardize individual ascent assembly modules and to automate their design process Frank et al. (2014). As a result, the task of automating the design of cost-optimal CAA-Structures has itself become a feasible undertaking since it can be regarded as a search for a 3D “skeleton” that indicates which ascent assembly modules are required and where they should be placed in order to ensure that the CAA-Structure provides access with minimal costs.

In spite of the apparent simplicity suggested by a 3D “skeleton”, there is still a large set of particularities and uncertainties associated with real-life CAA-Structure design tasks in modern engineer-to-order environments. In order to have a relevant but accessible formulation for analyzing and comparing various proofs of concept, in Section “[Description of Model Abstraction](#)” we introduce a 2D model abstraction of the optimal design task. Domain experts validated the results we obtained for a real-life industrial design scenario (described in Section “[Experimental Setup](#)”), thus indicating that the cost-optimal and innovative solutions obtained via the introduced 2D model abstraction and the proposed optimization strategy (described in Section “[Optimization Procedure](#)”) are a large step forward towards the final goal of fully automating the design of cost-optimal CAA-Structures.

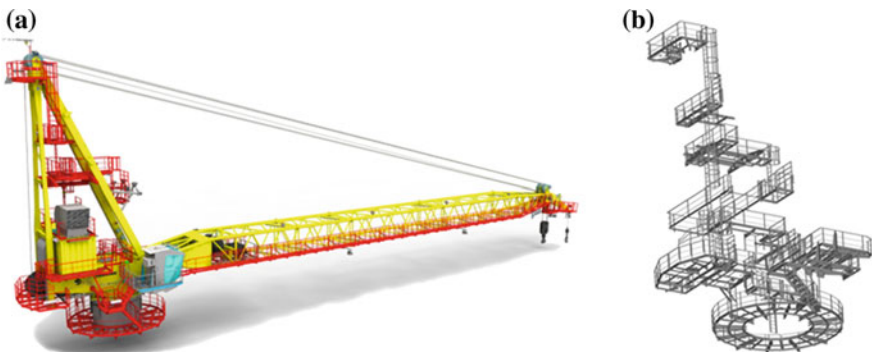


Fig. 1 An example of an offshore crane where different ascent assembly modules highlighted in red (a) are combined to form a fairly complex CAA-Structure (b)

Modelling and Formal Problem Statement

Description of Model Abstraction

A user that wishes to generate a cost-optimal CAA-Structure is expected to provide at least three inputs: a 3D model of the solid base/support object to which the CAA-Structure is to be attached, a set of desired points of access on this structure and information regarding potential obstacles that are defined on the 3D solid base object. The latter requirement is extremely relevant as obstacles indicate severe restrictions regarding the placement of ascent assembly modules in certain areas.

For example, in Fig. 2a we illustrate a simplified design case that involves a cuboid structure, five access points and four obstacle areas that are spread across three faces of the cuboid. A far clearer representation of this academic automated design scenario can be obtained by unfolding the 3D model. The result of the unfolding procedure, shown in Fig. 2b, is a 2D design surface that is characterized by a left edge - right edge continuity – i.e., line segments exiting the left edge at a certain height and orientation, should enter the right edge at the same height and orientation in order to model the circular structure of the facade. More importantly, as a result of the unfolding, the task of finding a cost-optimal 3D “skeleton” of the ascent assembly is transformed into that of discovering a simpler 2D design “skeleton”: a *cost-optimal 2D path network layout* on the 2D design surface that links all the points of interest while avoiding the obstacle areas.

Although an obvious simplification of the 3D case, we shall see in the next section that the resulting *2D path network layout problem* is not trivial.

Formal Definition of the Path Network Layout Problem

When considering a set of n user defined *access points/definition vertices* $\{p_1, \dots, p_n\}$, the goal of the 2D optimal path network layout problem is to discover a

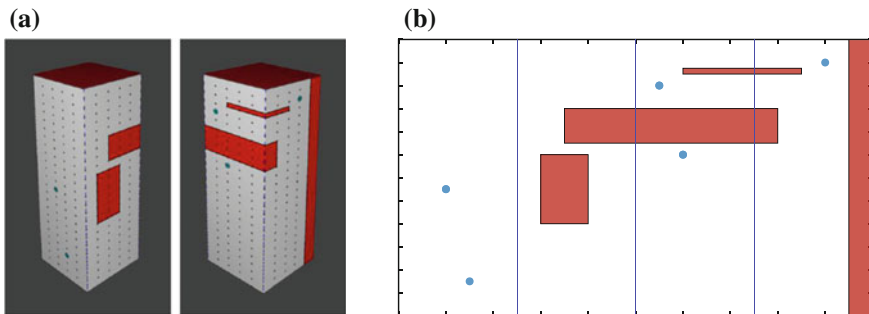


Fig. 2 A 3D model and the corresponding 2D design plane obtained after unfolding. Access points are marked with blue circles and obstacles are marked with red

(graph) structure T of minimal cost that links all these points. T must obviously span all the access points but it may also contain up to k well-placed extra points (2D vertices) $\{s_1, \dots, s_k\}$ that help minimize the total cost of T . Thus, E , the set of possible edges that contains all the segments that can be used to construct T , is defined over the union $\{p_1, \dots, p_n\} \cup \{s_1, \dots, s_k\}$. When considering a positive cost for connecting any two points, it is obvious that T is in fact a tree. Formally, the resulting minimal path optimization task can be defined as: *Determine $k \in \mathbb{N}$ and $s_1, \dots, s_k \in \mathbb{R} \times \mathbb{R}$ in order to minimize*

$$f_1(p_1, \dots, p_n, s_1, \dots, s_k) = \sum_{(ij) \in E} c(i, j)x_{(ij)}, \tag{1}$$

subject to:

$$\begin{aligned} x_{(ij)} &\in \{0, 1\}, \forall (ij) \in E \text{ and} \\ \sum_{(ij) \in E} x_{ij} &= (n + k) - 1 \text{ and} \\ \sum_{(ij) \in E, i \in F, j \in F} x_{ij} &\leq |F| - 1, \forall F \subseteq \{p_1, \dots, p_n, s_1, \dots, s_k\}, \end{aligned}$$

where $G = (\{p_1, \dots, p_n, s_1, \dots, s_k\}, E)$ is a complete graph.

The function $c(i, j)$ from Eq. (1) denotes the cost of linking vertices i and j . In the case of ascent assemblies, this cost can be defined as the combined price of individual modules (i.e., platform, stair, and ladder segments) and of connecting them (e.g., welding) required to construct a walkway between points i and j . While it is expected that, in the general case, $c(i, j)$ is proportional to the Euclidean distance between the two vertices, in more realistic scenarios, obstacles and other penalties do influence the cost function.

For example, when considering a slightly more realistic description of optimal layouts for CAA-Structures, one would likely consider inside $c(i, j)$ a large penalty $\Gamma_{(ij)}$ for assembly modules that extend into obstacle areas when connecting vertices i and j and another smaller penalty for assembly modules that are not placed at a preset angle requirement – e.g., platforms should be placed at an angle of exactly 0° to the horizontal axis, stairs at 45° , and ladders at 90° . All these “feasible/preferred” design angles should be provided as a user defined set, e.g., $U = \{0, 45, 90\}$. The resulting angle-aware cost function could be defined as:

$$c(i, j) = dist(i, j) \left(1 + \frac{\min B_{(ij)}}{100} z \right) + \Gamma_{(ij)} \tag{2}$$

where z is a parameter ($0 \leq z \leq 4$) that controls the magnitude of the angle penalty and $B_{(ij)}$ is a set that contains the absolute differences between $\alpha_{(ij)}$ – the horizontal angle of the segment (ij) – and the feasible placement angles stored in U . For instance, given the example in the previous paragraph, $B_{(ij)} = \{|\alpha_{(ij)} - 0|, |\alpha_{(ij)} -$

45], $|\alpha_{(ij)} - 90|$ }. For the set of tests we report over in the present study, $dist(i, j)$ marks the 2D Euclidean distance between vertices i and j .

It is noteworthy to remark that when $z = 0$ in Eq. (2) and one does not consider obstacle areas and a left-right continuity of the design plane, $c(i, j)$ is reduced to the Euclidean distance and Eq. (1) gives the definition of the well-known *Euclidean Steiner Tree Problem* (ESTP) Gilbert and Pollak (1968). Although they represent the simplest cases of the 2D path network layout problems we aim to solve, ESTPs are proven to be NP-hard Garey et al. (1977). Nevertheless, ESTPs have also been intensively studied by mathematicians and computer scientists and this opens up the possibility to compare (in part) our proposed solving strategy with other results from literature on standard benchmarks. In the context of ESTPs, the k points that help minimize the 2D path layout between the access points are called *Steiner points* and throughout this work we shall also maintain this naming in the context of optimal path network layouts for CAA-Structures. Furthermore, the NP-hard nature of ESTPs also motivates our strong preference for a metaheuristic-based solver. As such, we would like to inform the reader that the lexicon throughout the remainder of this work is tailored for the field of evolutionary computation De Jong (2006) – one of the most (historically) successful global optimization paradigms for tackling complicated optimization problems.

Finally, in the context of path network optimization tasks for ascent assemblies, opting for a value of $z = 0$ in Eq. (2) would result in a problem definition that enables the optimizer to freely explore the design space and quite possibly discover innovative designs (i.e., innovative ways of connecting the desired access points). However, the best results of such an “open” definition would (likely) only be interpreted as optimal design “suggestions” as building them to specification would be unfeasible. When opting for a larger value of the penalty parameter z and a realistic list of standard feasible angles, good results of the more “restricted” path optimization problem are far more likely to resemble “blueprints” of the ascent assembly.

Optimization Procedure

Solution Codification

A very important aspect of trying to solve the problem described in Section “[Experimental Setup](#)” is represented by the encoding of individuals/candidate solutions. First and foremost, a good encoding should be simple (*general*) in order to be compatible with many fitness assessment strategies and in order to allow for an immediate extension to 3D scenarios. Secondly, the encoding should also be *flexible* as the number of Steiner points required by each problem is unknown. Although the latter characteristic seems to hint towards a variable-length encoding, we argue in favour of a fixed-length variant in which the maximal number of the Steiner points expected to be discovered (i.e., k^*) is preset at a sufficiently large level. For example:

- in the case ESTPs, one can use the mathematically proven Gilbert and Pollak (1968) upper bound $k^* = n - 2$
- in the case of all the ascent assembly optimization problems presented in Section “Experimental Setup” we experimented with several settings in the range $n \leq k^* \leq 3n$.

The task of “deciding” the exact number of Steiner points required for solving the problem at hand is “passed” to the fitness assessment method described in the next section. Apart from the extra simplicity that enables the usage of various standard genetic operators, our choice for a fixed-length encoding is also motivated by the desire to counteract potential solution bloating - a well-known and harmful phenomenon in terms of both solution quality and convergence speed that is associated in the field of evolutionary computation (genetic programming in particular) with combinations of strong (evolutionary) selection pressure and variable-length encodings Langdon and Poli (1998).

After opting for fixed-length encodings, we adopted a basic real-valued representation $\mathbf{x} = (x_1, x_2, \dots, x_{2k^*-1}, x_{2k^*})$ of potential Steiner points, with the understanding that, given the encoded vertex $v_{(i,\mathbf{x})}$, $1 \leq i \leq k^*$, x_{2i-1} denotes the horizontal coordinate of $v_{(i,\mathbf{x})}$ and x_{2i} the vertical one.

The minimum and maximum ranges for $x_i \in \mathbf{x}$ are set according to the design scenario definition limits in the case of CAA-Structures and to the extreme coordinate values of the definition points in the case of ESTPs.

Fitness Assessment

Let $o_1(\mathbf{x})$ denote the ability of the vertices encoded in a given candidate solution \mathbf{x} to minimize Eq. (1). In order to estimate $o_1(\mathbf{x})$, we employ a two-step process:

- Firstly, we build the union between all the k^* vertices encoded by \mathbf{x} and the n definition points of the optimization scenario: $S_{\mathbf{x}} = \{v_{(1,\mathbf{x})}, \dots, v_{(k^*,\mathbf{x})}\} \cup \{p_1, \dots, p_n\}$.
- Secondly, starting with p_1 , we apply Prim’s algorithm Prim (1957) in order to construct $MT_{n,\mathbf{x}}$ – the *partial minimum spanning tree* (MST) over the set $S_{\mathbf{x}}$ that contains all n definition points. $MT_{n,\mathbf{x}}$ is a partial MST because the construction process is interrupted once all the definition points have been added to the tree.

Any vertex encoded in \mathbf{x} that remains unlinked by $MT_{n,\mathbf{x}}$ has the property that its placement is highly likely not to improve in any way the formation of an optimal-cost path between all the definition points $\{p_1, \dots, p_n\}$ – i.e., this vertex is deemed as having a low chance of being a potential Steiner point. We mark with $s_{i,\mathbf{x}}$, $i \in \{1, \dots, m\}$, $m \leq k^*$ the vertices encoded in \mathbf{x} that are part of $MT_{n,\mathbf{x}}$. Compared with the unlinked vertices, any $s_{i,\mathbf{x}}$ has a better chance of being useful in constructing an optimal path between the definition points and is thus deemed a *potential Steiner point*. When considering previous notations, and denoting with $\Phi(MT_{n,\mathbf{x}})$ the total cost associated with $MT_{n,\mathbf{x}}$, we have that $\Phi(MT_{n,\mathbf{x}}) = f_1(p_1, \dots, p_n, s_{1,\mathbf{x}}, \dots, s_{m,\mathbf{x}})$ where function f_1 is defined in Eq. (1).

We argue that $o_1(\mathbf{x})$ can be well approximated by $\Phi(MT_{n,\mathbf{x}})$ because the closer the set of potential Steiner points in \mathbf{x} is to $\{s_1, \dots, s_k\}$, i.e., to the actual Steiner point set that represents the solution to Eq. (1), the closer $f_1(p_1, \dots, p_n, s_{i,\mathbf{x}}, \dots, s_{m,\mathbf{x}})$ is to $f_1(p_1, \dots, p_n, s_1, \dots, s_k)$.

When considering the main motivation behind the present work (i.e., optimizing real-life industrial designs), it is highly likely that a more advanced future model abstraction might yield *secondary requirements regarding optimality*. For instance, these requirements might relate to:

1. the complexity of the overall CAA-Structure design (i.e., number of different module types that is required),
2. ensuring different levels of ease-of-access for different definition points,
3. CAA-Structure building time given present stocks of individual ascent assembly modules.

Such possible secondary requirements appear to be rather conflicting with the currently identified primary one (i.e., cost minimization) and modeling them via penalties and rewards is expected to be extremely cumbersome. Alternatively, formalizing them as optimization objectives in their own right would be more natural and should yield better results from the perspective of a decision maker.

Motivated largely by the previous considerations but also by initial attempts to optimize $o_1(\mathbf{x})$ on benchmark ESTPs using evolutionary algorithms that were less successful than anticipated (showing signs of premature convergence), we defined an artificial secondary objective $o_2(\mathbf{x})$. This second objective is designed to be (slightly) conflicting with o_1 and is defined as:

$$o_2(\mathbf{x}) = \frac{1}{n-1} \left(\sum_{r=2}^n impr_{MST(r)} * 2^{size_{MST(r)}} \right) - 1.1^m, \tag{3}$$

where:

$$impr_{MST(r)} = \Phi(MT_{r,\mathbf{x}}) - \Phi(MST_r) \text{ and}$$

$$size_{MST(r)} = 3 - \frac{3\Phi(MST_r)}{\Phi(MST_n)}.$$

Inside Eq. (3), when considering that p_r is the r^{th} user defined access point, in an analogous way to $\Phi(MT_{n,\mathbf{x}})$, we have that:

- $\Phi(MST_r)$ is the cost of the minimum spanning tree constructed over the set $\{p_1, \dots, p_r\}$, i.e., $\Phi(MST_r) = f_1(p_1, \dots, p_r)$;
- $\Phi(MT_{r,\mathbf{x}})$ is the total cost of the *partial minimal spanning tree* constructed over the union $S_{\mathbf{x}} = \{v_{(1,\mathbf{x})}, \dots, v_{(k^*,\mathbf{x})}\} \cup \{p_1, \dots, p_r\}$

This means that $o_2(\mathbf{x})$ computes the average level to which \mathbf{x} is able to solve Eq. (1) for every incremental subset of definition points that is obtained when constructing the minimal spanning tree over $\{p_1, \dots, p_n\}$. The smaller the subset the more important it is weighted inside the average and there is a small bonus for candidate solutions that achieve good results with a reduced number m of potential Steiner points.

Finally we have chosen to solve a multi-objective optimization problem that aims to simultaneously minimize both $o_1(\mathbf{x})$ and $o_2(\mathbf{x})$. Although empirically validated by all the results presented in Section “Results”, the inclusion of $o_2(\mathbf{x})$ alongside the main path minimization objective is highly counter-intuitive. The reason for this decision is two-fold:

- $o_2(\mathbf{x})$ is engineered to induce both some level of niching during the evolutionary search as well as a biasing of the multi-objective search towards robust candidate solutions that encode potential Steiner points which are generically well placed – i.e., able to improve the total minimal path in key locations that are common to many sub-paths.
- Having a multi-objective formulation enables us to check whether our assumption that $\Phi(MT_{n,x})$ is a good enough approximation for $o_1(\mathbf{x})$ also holds when faced with a conflicting optimization objective that, to a certain extent, steers the search towards path layouts that are not necessarily cost-optimal.

The Multi-objective Solver

In order to solve the previously introduced multi-objective optimization (MOO) problem, we performed a limited set of initial tests with NSGA-II Deb et al. (2002) – a classical multi-objective evolutionary algorithm (MOEA) – and with DECMO2 Zăvoianu et al. (2014). The latter is a newer hybrid and adaptive evolutionary approach specially designed for rapid convergence on a wide class of problems. DECMO2 was designed to capitalize on previous insights Zăvoianu et al. (2013) that a cooperative coevolutionary strategy can deliver very competitive results on a wide range of MOO problems. As DECMO2 exhibited a better balance between convergence speed and final solution quality, we adopted it as our default solver. The DECMO2 evolutionary model is presented in Algorithm 1 and its main feature is that it effectively integrates three different MOO search space exploration paradigms.

Firstly, P , one of the two equally-sized coevolved subpopulations in DECMO2 implements a SPEA2 Zitzler et al. (2002) evolutionary model that is based on *environmental selection* (notation E_{sel}) - a selection for survival mechanism based on Pareto dominance as a primary metric and a crowding distance in objective space as a secondary metric. Apart from E_{sel} , this evolutionary paradigm also relies on the simulated binary crossover (SBX) Deb and Agrawal (1995) and polynomial mutation (PM) Deb (2001) genetic operators. It is noteworthy that inside E_{sel} , DECMO2 uses a slightly modified version of environmental selection that filters objective-wise duplicates from the returned solution set.

Secondly, Q – the other coevolved subpopulation – implements a GDE3-like Kukkonen and Lampinen (2005) search behaviour that focuses on exploiting the very good performance of the differential evolution paradigm Storn and Price (1997) on continuous optimization problems.

Algorithm 1 The DECMO2 Zăvoianu et al. (2014) evolutionary model

```

1: function DECMO2(problem, asize, maxGen)
2:    $P, Q \leftarrow \emptyset$ 
3:    $\langle p_{size}, q_{size}, e_{size} \rangle \leftarrow \text{COMPUTESIZES}(a_{size})$ 
4:    $A \leftarrow \text{INITIALIZEARCHIVE}(\textit{problem}, a_{size})$ 
5:    $i \leftarrow 1$ 
6:   while  $i \leq a_{size}$  do
7:      $\mathbf{x} \leftarrow \text{CREATEINDIVIDUAL}(\textit{problem})$ 
8:      $A \leftarrow \text{INSERTINTOARCHIVE}(A, \mathbf{x})$ 
9:     if  $i \leq p_{size}$  then
10:        $P \leftarrow P \cup \{\mathbf{x}\}$ 
11:     else
12:       if  $i \leq (p_{size} + q_{size})$  and  $i > p_{size}$  then
13:          $Q \leftarrow Q \cup \{\mathbf{x}\}$ 
14:       end if
15:     end if
16:      $i \leftarrow i + 1$ 
17:   end while
18:    $\phi^P, \phi^Q, \phi^A, t \leftarrow 1$ 
19:   while  $t \leq \textit{maxGen}$  do
20:      $p_{bonus}, q_{bonus} \leftarrow 0$ 
21:      $a_{bonus} \leftarrow e_{size}$ 
22:     if  $t \in \{2k + 1 : k \in \mathbb{Z}\}$  then
23:        $p_{bonus}, q_{bonus}, a_{bonus} \leftarrow 0$ 
24:       if  $\phi^P > \phi^Q$  and  $\phi^P > \phi^A$  then
25:          $p_{bonus} \leftarrow e_{size} \wedge a_{bonus} \leftarrow 0$ 
26:       end if
27:       if  $\phi^Q > \phi^P$  and  $\phi^Q > \phi^A$  then
28:          $q_{bonus} \leftarrow e_{size} \wedge a_{bonus} \leftarrow 0$ 
29:       end if
30:     end if
31:      $\langle P, \phi^P \rangle \leftarrow \text{EVOGENSPEA2}(P, p_{size} + p_{bonus})$ 
32:      $\langle Q, \phi^Q \rangle \leftarrow \text{EVOGENDE}(Q, q_{size} + q_{bonus})$ 
33:      $\phi^A \leftarrow \text{EVODIRARCHIVEIND}(A, a_{bonus})$ 
34:      $E \leftarrow E_{sel}(P \cup Q \cup A, e_{size})$ 
35:      $P \leftarrow E_{sel}(P \cup E, p_{size})$ 
36:      $Q \leftarrow E_{sel}(Q \cup E, q_{size})$ 
37:      $t \leftarrow t + 1$ 
38:   end while
39:   return  $E_{sel}(P \cup Q \cup A, a_{size})$ 
40: end function

```

Thirdly, the last MOO paradigm incorporated in DECMO2 comes in the form of an archive of well-spaced elite solutions, A , that is maintained according to a decomposition-based principle similar to the one popularized by MOEA/D-DE Zhang et al. (2009). Even though at certain times a limited number of new individuals is evolved directly from the archive, the main purpose of A is to preserve an accurate approximation of the Pareto front.

DECMO2 is also designed to dynamically pivot towards the evolutionary paradigm that was more successful during the latest stage of the run by allowing the part of

the algorithm that implements this paradigm to generate a total of $e_{size} = \frac{2}{9}|P|$ individuals more than usual. This means that during the run, a performance bonus is awarded based on perceived current space-exploration performance.

Experimental Setup

DECMO2 Parameterization

For all the numerical experiments that we report on, we used a total population size of $a_{size} = 400$ for DECMO2 and the literature recommended parameter settings for the coevolved subpopulations of the solver. Thus, for subpopulation P of size 180, we used a value of 0.9 for the crossover probability and 20 for the crossover distribution index of the SBX operator and a value of $1/|x|$ for the mutation probability and 20 for the mutation distribution index of the PM operator. Subpopulation Q was evolved according to a DE/rand/1/bin strategy Storn and Price (1997) in which the control parameter F was set at 0.5 and the crossover factor CR was set at 0.3. Spacing inside the archive A was maintained via a weighted Tschebyscheff distance measure.

We evaluated 100.000 solution candidates during each optimization run (i.e., $maxGen = 250$) and we report on the best result out of 3 repeats for each numerical experiment. Since the secondary objective of our MOOP is an artificial placeholder that has no practical importance when assessing the overall result of the optimization, we always only report the best discovered solution with regard to $o_1(x)$.

Benchmark Problems and Industrial Test Case

In order to demonstrate the ability of our approach, we compared the results obtained by DECMO2 on 15 problems from a benchmark ESTP set Soukup and Chow (1973) with those of two reference solvers: one based on a geometrically motivated heuristic Beasley (1992) and another one that uses artificial neural networks Bhaumik (1994).

For initial insight on how our method performs on optimization scenarios that are more representative for the ascent assembly domain, we applied DECMO2 on 4 academic test cases: the one illustrated at the beginning of this paper (A1) in Section “[Description of Model Abstraction](#)” and three more derivations (A2, A3, and A4) based on the more challenging access point placement from problem no. 12 of the ESTP benchmark set. On all these tests we used the setting $z = 0$ to parameterize the cost function from Eq. (2) and thus optimize for the minimum Euclidean distance.

The most realistic cost-optimal CAA-Structure design scenario we investigated was proposed by Liebherr-Werk Nenzing GmbH (LWN) LWN (2017) – a manufacturer of a wide range of products including various types of cranes. More specifically, we investigated cost-optimal CAA-Structures that allow access to user-specified re-

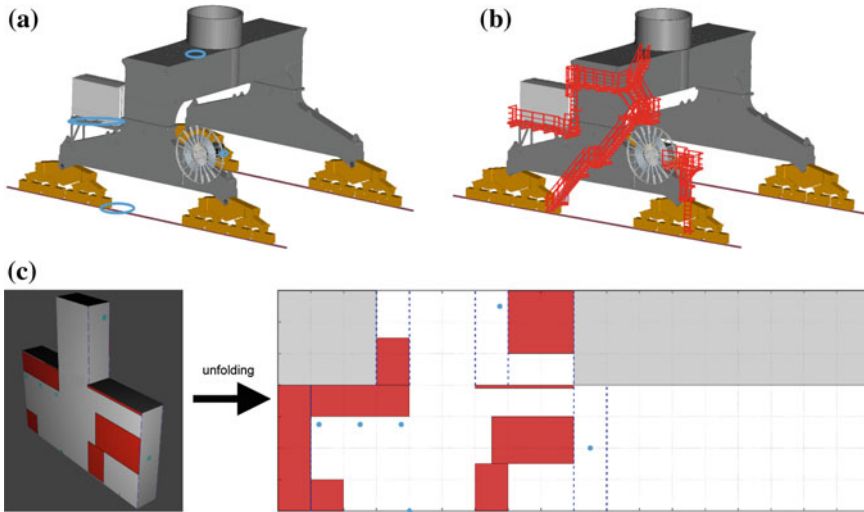


Fig. 3 A CAD model of the gantry of a Liebherr mobile harbour crane with highlighted user-specified access points (a) expert-designed ascent assembly solution (b) and complementary 3D and unfolded 2D model abstraction (c)

gions of interest on the gantry of a mobile harbour crane (please see Fig. 3a). Figure 3b presents an expert-designed CAA-Structure attached to the gantry and we aim to use the unfolding-based 2D model abstraction from Fig. 3c (obtained by vertically stacking two cuboids) to explore complementary optimal designs that might provide interesting insights to LWN.

In order to provide a balance between innovative and realistic cost-optimal solutions for CAA-Structures, we considered two test case variations (TC1 and TC2) in which the ground access point is placed at different positions along the horizontal axis and four different cost settings:

- C1 — The first cost setting uses a value of $z = 0$ to parameterize the cost function from Eq. (2). Given the infinite degrees of freedom, optimal designs discovered for this setting are expected to have the smallest total path network (Euclidean) distance and can be used as a generic structural reference when assessing more constrained cost-optimal designs.
- C2 — The second cost setting uses a value of $z = 4$ and a list of preferred design angles $U = \{0, 45, 90\}$ and aims to deliver ascent assembly designs that only use the three standard assembly components: horizontal platforms, stairs and vertical ladders.
- C3 — The third cost setting uses a softer angle-wise constraint factor of $z = 1$ and a minimal list of preferred design angles $U = \{0, 90\}$ and aims to deliver ascent assembly designs that have a real-life minimal cost as they only use horizontal platforms and vertical ladders.

- C4 — The fourth cost setting uses a value of $z = 4$ and a minimal list of preferred design angles $U = \{0, 30, 45\}$ and aims to deliver CAA-Structures that offer a higher degree of access (no mandatory use of hands) by only using platform modules and two types of staircase modules (mild and regular inclination).

Results

Performance on Artificial Problems

The comparative performance on benchmark ESTPs is presented in Table 1 and indicates that our solving strategy based on DECMO2 and a MOOP formulation is very competitive for ESTP instances that have a low-to-medium number of definition (access) points.

The results for the four academic CAA-Structure design scenarios are presented in Fig. 4 and they indicate that the DECMO2-based solving strategy is quite general and able to both efficiently avoid obstacles as well as profit from the left edge - right edge continuity.

Table 1 Comparative performance of DECMO2 on ESTPs. Best results are highlighted and * marks problems with an unknown optimum

Problem Id. Soukup and Chow (1973)	n	Minimum Euclidean Steiner tree			
		Baseline Soukup and Chow (1973)	Beasley (1992)	Bhaumik (1994)	DECMO2
1	5	1.6644	1.6644	1.6644	1.6644
2B	8	2.1387	2.1387	2.1393	2.1387
2D	12	2.2223	2.1842	2.2979	2.1842
2G*	7	1.5878	1.6018	1.7019	1.5594
3	6	1.6472	1.5988	1.6553	1.5988
6	9	1.2733	1.2862	1.3024	1.2733
11*	64	3.8513	3.8380	3.9707	3.8274
12*	14	1.7222	1.7222	1.7989	1.7067
15A	5	0.5130	0.5130	0.5236	0.5130
18*	12	1.0332	1.0421	1.0782	1.0241
19B*	19	2.8567	2.8408	2.9689	2.8286
26*	20	1.9767	2.2770	1.9785	1.9785
28*	16	2.3671	2.3446	2.4048	2.3309
29*	17	2.1974	2.1974	2.2076	2.1869
31*	16	1.4220	1.3999	1.4343	1.3660

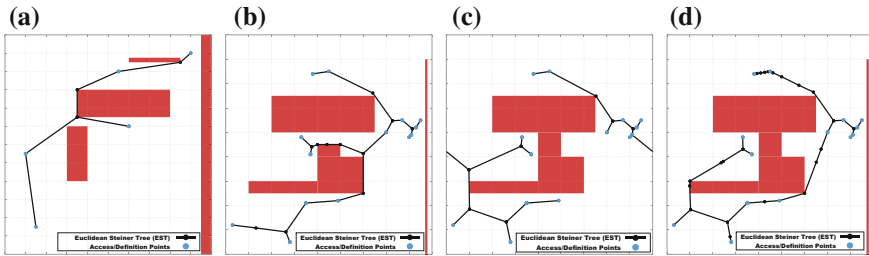


Fig. 4 Results for the four academic test cases

Results for the LWN Industrial Test Case

The best solutions discovered for the two test case variants of the Liebherr mobile harbour crane scenario when considering the four different cost settings are plotted in Figs. 5 and 6.

A visual inspection of the results for each test case reveals that imposing angle-wise restrictions on the overall design of the ascent assembly can be successfully accommodated by the DECMO2-based optimization strategy. Furthermore, while angle-wise restrictions do influence the optimization outcome, the generic (star-shaped) structure that characterizes the expert CAA-Structure design is confirmed by all the cost-optimal results obtained for this somewhat simplistic test case.

As specific observations related to the discovered optimal CAA-Structure designs, it is noteworthy that:

- The less restrictive setting $z = 1$ in C3 can result in designs that apart from platforms and ladder segments also feature ramps as shown in Fig. 5c.
- The cost setting C4 seems to result in designs (e.g., Fig. 6d) that resemble more closely the expert-designed CAA-Structure illustrated in Fig. 3b. This indicates that accounting for ease-of-access concerns should be enforced in future extensions of the model abstraction.
- Shifting the bottom access point along the horizontal axis induces a small and local effect when unlimited degrees of freedom are allowed (Figs. 5a and 6a) but the effect on the overall design of assembly is larger and global when considering angle-wise restrictions. This indicates that using a continuous problem formulation and a limited set of domain-dependent restrictions can yield innovative optimal design suggestions that an expert might overlook.

General Conclusions and Outlook

In the present work we have introduced an initial, practical model abstraction for the task of automating the *cost-optimal design of complete ascent assembly structures*. In order to tackle in a domain-realistic manner the *2D Path network layout problem*

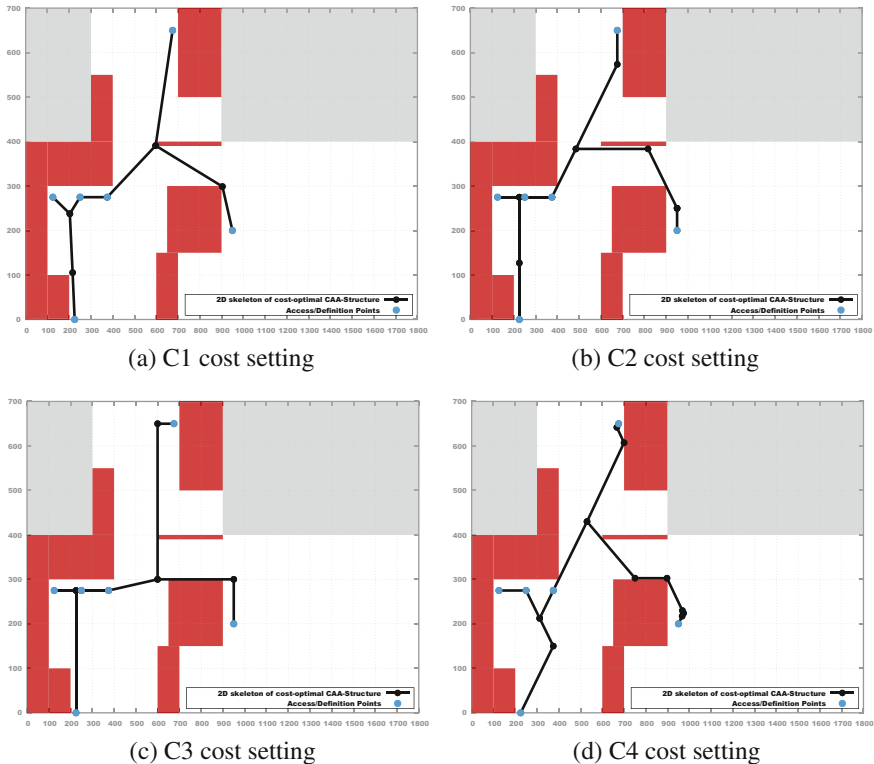


Fig. 5 CAA-Structure optimization results for the LWN TC1 optimization scenario using different cost functions

that emerges from the aforementioned model abstraction, we propose an optimization procedure based on a multi-objective problem formulation and an advanced coevolution-based solver – DECMO2. As results obtained on benchmark and academic test cases were very encouraging, we also applied our approach on a real-life CAA-Structure design scenario provided by an industrial partner.

The results for the real-life CAA-Structure optimization scenario also empirically support the validity of our approach. Thus, by employing appropriate cost functions, formalizing the CAA-Structure optimization problem on a continuous design space facilitates the discovery of a wide range of innovative designs that provide design engineers with valuable insight regarding the trade-offs between the best theoretical CAA-Structure design (that requires infinite degrees of freedom and new types of assembly modules) and the best practical CAA-Structure design that only requires traditionally used ascent assembly modules.

In the future we plan to investigate the hybridization potential between our current approach and a complementary design strategy Hellwig et al. (2017) that is based on a discretization of the design surface and that delivers competitive results on CAA-Structure optimization scenarios with strong angle-wise restrictions.

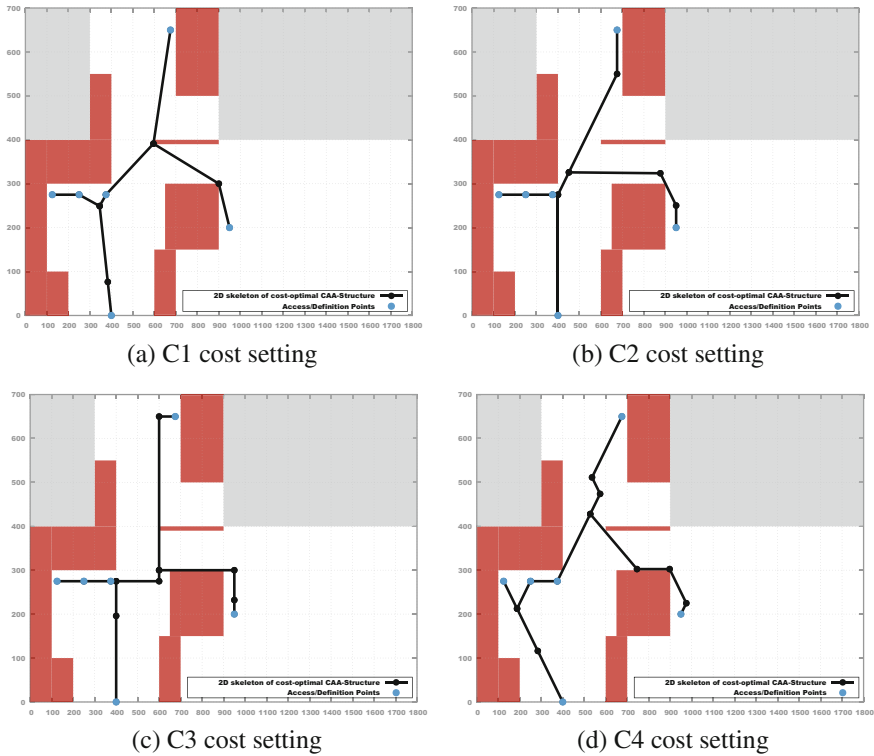


Fig. 6 CAA-Structure optimization results for the LWN TC2 optimization scenario using different cost functions

Since the search logic of our DECMO2-based solving strategy is very loosely bound to the 2D model abstraction, future work will also revolve around solving the cost-optimal design problems directly in 3D space. The reason is that the simple 2D representation is rather restrictive for several real world applications. For example, it is not possible to model the crane from Fig. 1a with it.

Acknowledgements This work was supported by the K-Project “Advanced Engineering Design Automation” (AEDA) that is financed under the COMET funding scheme of the Austrian Research Promotion Agency.

References

Beasley JE (1992) A heuristic for Euclidean and rectilinear Steiner problems. *Eur J Oper Res* 58(2):284–292
Bhaumik B (1994) A neural network for the Steiner minimal tree problem. *Biol Cyber* 70(5):485–494

- De Jong KA (2006) *Evolutionary computation: a unified approach*. MIT press, Cambridge, MA
- Deb K (2001) *Multi-objective optimization using evolutionary algorithms*. Wiley, New York
- Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9:115–148
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
- Frank G, Entner D, Prante T, Khachatouri V, Schwarz M (2014) Towards a generic framework of engineering design automation for creating complex CAD models. *Int J Adv Syst Meas* 7(1):179–192
- Garey MR, Graham RL, Johnson DS (1977) The complexity of computing Steiner minimal trees. *SIAM J Appl Math* 32(4):835–859
- Gilbert E, Pollak H (1968) Steiner minimal trees. *SIAM J Appl Math* 16(1):1–29
- Hellwig M, Entner D, Prante T, Zăvoianu AC, Schwarz M, Fink K (2017) Optimization of ascent assembly design based on a combinatorial problem representation. In: *Proceedings of EUROGEN 2017*, Madrid, Spain, September 13–15
- Kukkonen S, Lampinen J (2005) GDE3: The third evolution step of generalized differential evolution. In: *IEEE Congress on evolutionary computation (CEC 2005)*. IEEE Press, New York, pp 443–450
- Langdon WB, Poli R (1998) Fitness causes bloat. In: Chawdhry P et al (eds) *Soft computing in engineering design and manufacturing*. Springer, Berlin, pp 13–22
- LWN (2017) Liebherr-Werk Nenzing GmbH. <http://www.liebherr.com/en-GB/35267.wfw>. Accessed 06 March 2017
- Prim RC (1957) Shortest connection networks and some generalizations. *Bell Labs Tech J* 36(6):1389–1401
- Soukup J, Chow W (1973) Set of test problems for the minimum length connection networks. *ACM SIGMAP Bull* 15:48–51
- Storn R, Price KV (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J Global Optim* 11(4):341–359
- Zhang Q, Liu W, Li H (2009) The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. Technical Report, School of CS & EE, University of Essex
- Zitzler E, Laumanns M, Thiele L (2002) SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary methods for design, optimisation and control with application to industrial problems (EUROGEN 2001)*, International Center for Numerical Methods in Engineering (CIMNE), pp 95–100
- Zăvoianu AC, Lughofer E, Amrhein W, Klement EP (2013) Efficient multi-objective optimization using 2-population cooperative coevolution. In: *Computer aided systems theory - EUROCAST 2013*. Springer, Berlin, pp 251–258 (Lecture Notes in Computer Science)
- Zăvoianu AC, Lughofer E, Bramerdorfer G, Amrhein W, Klement EP (2014) DECMO2: a robust hybrid and adaptive multi-objective evolutionary algorithm. *Soft Comput* 19(12):3551–3569. <https://doi.org/10.1007/s00500-014-1308-7>

Taking Advantage of 3D Printing So as to Simultaneously Reduce Weight and Mechanical Bonding Stress



Markus Schatz, Robert Schweikle, Christian Lausch,
Michael Jentsch and Werner Konrad

Abstract 3D printing is recently gaining attention, yet only few researchers address underlying design principles such as minimal thickness to shape ratio; even though they are essential to industrial applications. The authors outline an optimization and verification approach considering structural aspects such as stiffness and strength as well as producibility and structural performance. This multitude of disciplines brought forth objectives, being diametral to each other. An example is giving by simultaneous mass reduction by increasing the part's strength performance. So as to harmonize all of those objectives to an optimal compromise, topology optimization has been used in tandem with consultations of design and structural experts. Additionally, an aerospace part, namely a 3D printed titanium insert was built and glued into an aluminium sandwich panel with carbon fiber reinforced plastic face sheets. This composite panel was then subjected to actual flight loads of the METImage satellite campaign. During all mechanical and thermal tests, cracks are captured via acoustic monitoring. Studying all test results revealed, that the approach brought forth multiple advances such as; reduced weight, increased mass-specific effective stiffness and lower mechanical bonding stresses, which increased overall structural strength.

Motivation and Introduction to Mechanical Frame of METImage

The part being optimized herein, is developed within the METImage project. METImage is one of the key instruments of the EUMETSAT Polar System satellite campaign; being designed to optically capture meteorological data such as cloud coverage and ground temperature. For more information on the mission and the instrument itself, consult (Wallner et al. 2016). All detectors and mechanism of that optical camera are accommodated in a cube as illustrated with Fig. 1.

M. Schatz (✉) · R. Schweikle · C. Lausch · M. Jentsch · W. Konrad
Airbus Defence & Space GmbH, Claude-Dornier-Strasse, Immenstaad, Germany
e-mail: schatz@llb.mw.tum.de

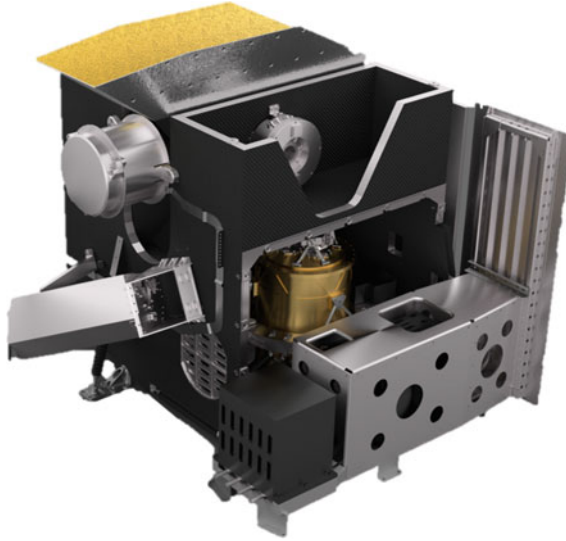


Fig. 1 Rendering of the optical instrument METImage, which later will be mounted on the EUMETSAT Polar System - Second Generation (EPS-SG)

This cube is supported by six carbon fiber reinforced polymer (CFRP) struts which evidently transmit all loads. The sandwich is comprised of carbon fiber reinforced plastic (CFRP) face sheets including stiffeners (top and bottom side), an aluminum honeycomb core and titanium inserts. So as to verify the design of that design, a smaller surrogate structure was developed as well. Figure 2 illustrates the sandwich structure designated to verify the METImage structural design and, moreover, proof

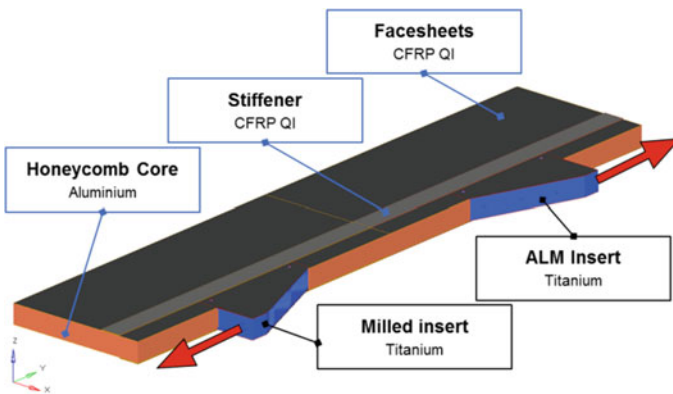


Fig. 2 The developed design verification structure; the panel insert bread board (PIB). This structure emulates the cubic sandwich structure of METImage

the applicability of 3D printed parts; or in other words Additive Layer Manufactured (ALM). To enable a comparison beside the 3D printed insert a conventional milled insert is integrated as well.

Literature Review

One of the first scientists, to study the most efficient allocation of material in structures was Michell (1904). His concepts relied on trusses and were of discrete nature: truss yes or no. Prager (1974) and Rozvany (1976) picked up on Michells work by finding and studying analytical solutions of topology optimization problems. A remarkable contribution was made by Bendsøe in the 90s (Bendsøe 1989). He and his researchers formalized topology optimization problems, where continuous variables were introduced so as to enable efficient numerical solution finding. Such that ultimately, a convergence to discrete structures is assured, those variables were penalized; the well-known SIMP approach. From there on, topology optimization became the topic of many research works as outlined by Eschenauer and Niels (2001). Since nowadays, where 3D printing is potent enough to manufacture nearing any topology in plastic as well as in metals (aluminum, titanium and even steel), the application of topology optimization experienced an unseen upswing in industry.

However, there are tricky spots. Sigmund and his colleagues actually revealed a blind spot, by showing how filters, degrees of freedoms (number of finite elements) and further numerical parameters have their imprint on topology optimization (Sigmund et al. 2016). They did so, by studying multiple structures with a tremendous number of finite elements, i.e. in the big millions. One example they gave, was a torsion loaded tube, where obviously a hollow axle performs best. The following Fig. 3 depicts their optimal result on the left, an axle and a sub-optimal design on the right. The sub-optimal design was derived with conventional filters and moderate number of elements. They underpinned therewith their statement, that topology optimization should be carefully interpreted and that the presence of Michell structures clearly is not a sufficient indicator for optimality. Furthermore, the strong design bias caused by classical domain filters and finite element models with an element count of an industry scale was proven.

Knowing this, the authors of this work, did set-up an approach, where topology optimization results are carefully interpreted in tandem with experts from different disciplines.

A Conventional Insert Design

A conventional design of such an insert is most frequently realized via milling. In order to ensure accessibility of the milling head while having closed surfaces such that the honeycomb core can be glued to the insert, this part has to be realized as

Fig. 3 One of the findings of Sigmund et al. (2016). On the right side, a conventional topology optimization result and on the left the real structural optimum

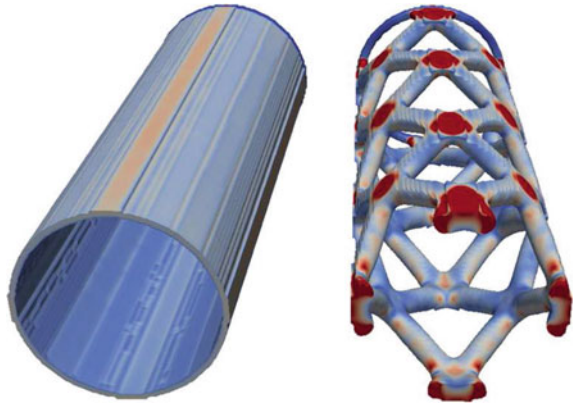
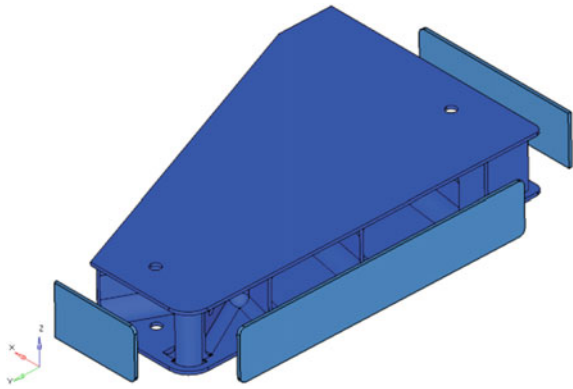


Fig. 4 Conventional design of a titanium sandwich insert. The three light blue plates represent so called back plates



follows. The insert itself is milled down to the outer extend defined by strength and stiffness requirements. For covering all faces, the so called back-plates are to be designed; light blue plates in Fig. 4. These plates are then glued to the inner structure; dark blue in Fig. 4.

This design however brings along several drawbacks. First of all, the back-plates translate to a direct mass increase, since they do not transmit loads nor considerably contribute to the overall stiffness. In addition, more glueing regions are introduced, which possibly could fail during thermal cycling, i.e. thermo-elastic incompatibilities. An even more drastic drawback, is the lack of design freedom in terms of wall thickness owed to the milling process. This is a disadvantage because, reducing the thickness around the edges would reduce stiffness jumps, thereby mitigating stress peaks. This was analyzed for the most critical loading scenario, a cooling down from 55 to -20°C , causing thermos-elastic deformations (TED) and stresses. This is due to incompatibilities of the individual materials thermal coefficient of expansion. The results of numerical analysis are given next with Fig. 5.

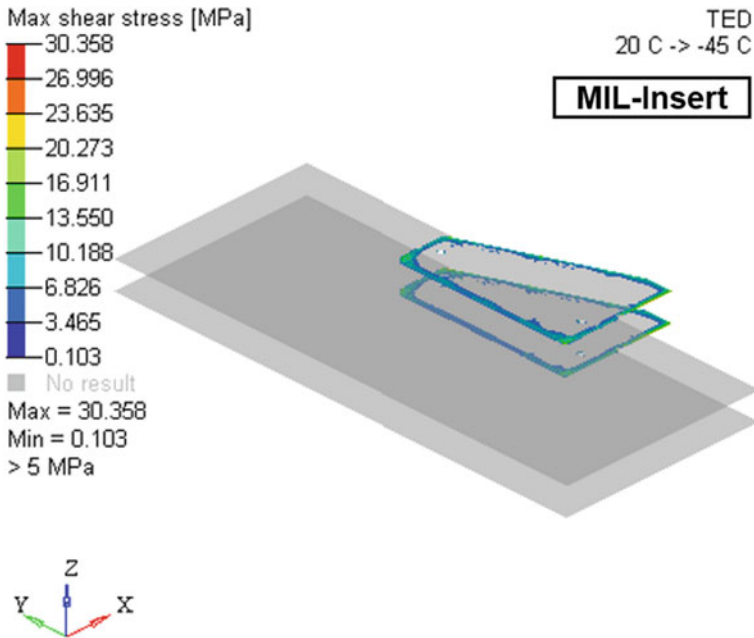


Fig. 5 Stresses in adhesive layer in case of the conventional insert design, i.e. milled

Thus one limiting factor for the ultimate load, certainly is the overshoot of stresses acting in those adhesive layers. For that sake, one major objective, among the minimization of mass and costs, was the reduction of stress peaks in that area.

The Design Process

In this section, the actual design process as developed and implemented at Airbus Defence & Space in Immenstaad is being introduced. As outlined in Section “Literature Review”, using FE models in concert with conventional filters, allowing to be solved with industrial means, topology optimization is likely to yield one dimensional structures. It will do so, even though shell-like structures would be superior. Honoring this, it was decided to use topology optimization just as one tool in a sequence of process steps. This is illustrated with the following Fig. 6. A general rationale of this design process is, that aiming for a topology optimization in one shot is not viable in current practice, but to acquire engineers expertise and rely on a robust interpretation.

A proper design task definition is evidently of paramount importance, which is why it marks the first step of our design process; *Problem Definition*. Thereafter, a *Topology Optimization* was set up for identifying optimal load transmission paths.

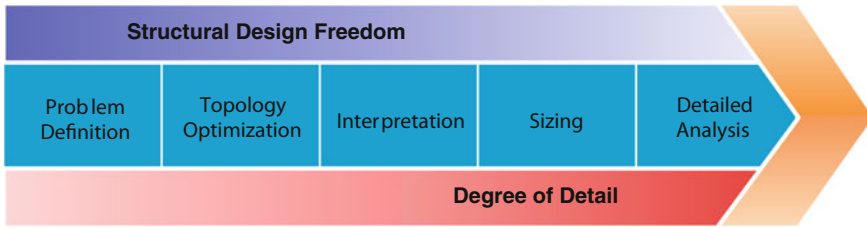


Fig. 6 Implemented design process

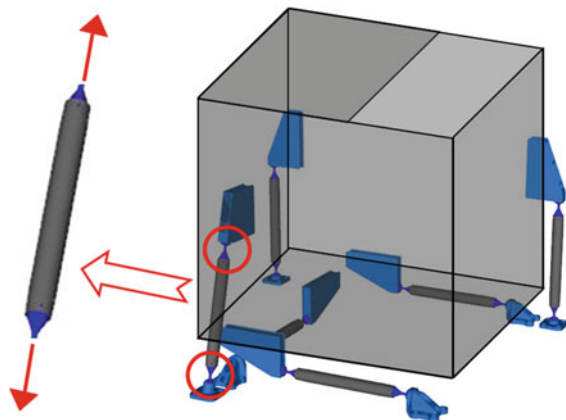
The subsequent step *Interpretation* addresses the translation of optimization results into technically viable designs. The most attractive design is then sized so as to further lift potentials. *Detailed numerical analyses* ensure desired design maturity through high fidelity simulations. All those steps are discussed in detail next.

Problem Definition

Vibrations of rocket engines, acoustic excitations and separation shocks form the design driving load spectra. A mechanical design limit load of 20kN acting on a single support strut, was derived for use in quasi-static analysis. Not only sustaining this load, but further efficiently transmitting this load into the composite sandwich structure, i.e. with moderate mass expanses and space, marks the challenging objective of the following work. 3D printing in concert with topology optimization and thorough mechanical reflecting cumulated in an attractive solution.

In the case of METimage, the actual mechanical loads are derivable with ease, since the instrument sandwich cube is iso-statically supported by six struts as mentioned in Section “[Motivation and Introduction to Mechanical Frame of METimage](#)”. Thus it is supported by six struts, where each strut has thin fittings at the end, such that they are able to bend almost as easy as a hinge. For that reason, struts evidently just transmit tensional loads as depicted in Fig. 7.

Fig. 7 Underlying iso-static strut concept of METimage and the derivable load transmission path



Topology Optimization

With Fig. 8 the FE model used for conducting the topology optimization is given. As depicted, it comprises the insert itself - clearly chosen to be the design space of the conducted optimization - and further, the sandwich core, face sheets et cetera so as to consider their stiffness. Tensile loading of 20 kN (red arrow) marks the loading scenario, whereby the overall stiffness was minimized. Through a variation of mass constraints multiple optima were found. One of the found optima is illustrated with Fig. 8.

Interpretation

As outlined above, multiple optimization runs were performed each yielding distinct results which together did point in the same direction in terms of mechanical transmission of the design limit load. For instance, it is obvious, that the main transmission path is to be designed as illustrated via the red arrow in Fig. 9. Since a certain robustness is more than desirable, the load should be smoothly distributed as highlighted by the yellow fan of arrows. Aside, this distribution moreover mitigates stress peaks which would arise in case of one single load transmission path.

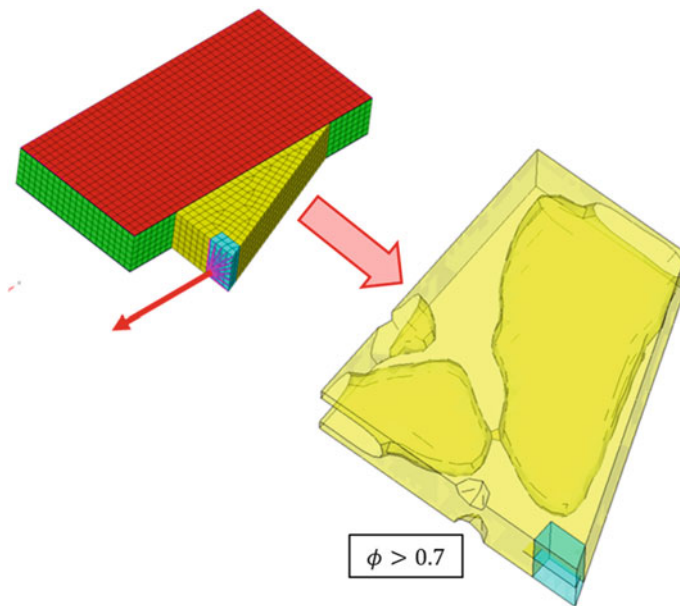
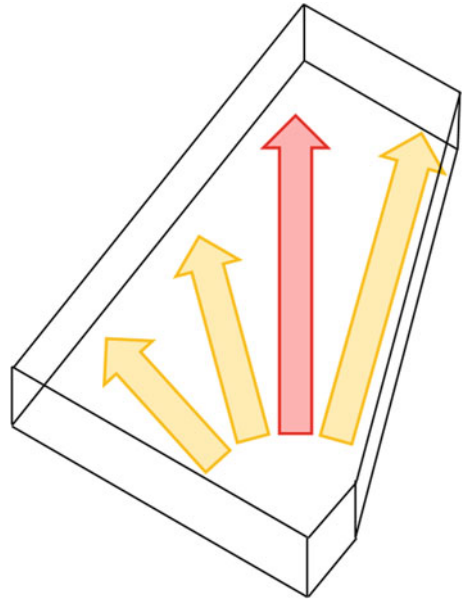


Fig. 8 The FE model used for the topology optimization, where only the yellow part did actually serve as design space is depicted in the back. One of the many topology optimization results. Only elements displaying an element density above 70% are highlighted

Fig. 9 Major and minor load transmission paths



Adding to the mechanical considerations, the following issues needed to be appropriately addressed during this interpretation phase as well:

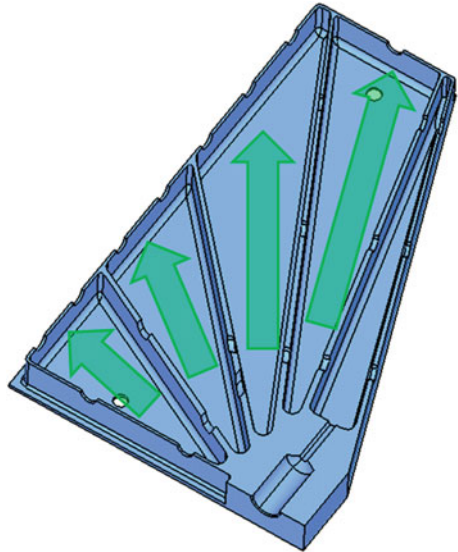
- Thermal incompatibilities, i.e. different coefficient of thermal expansion
- Limitations arising from 3D printing process
- Surface treatments for increased gluing performance
- Cleaness, i.e.
 - Accessibility of all surfaces being processed
 - Origin and kind of contaminants
- Out-gasing right after launch, e.g. trapped air in pockets

Because of all those aspects, the interpretation was realized by a sequence of consultations with experts and meetings for making arrangements. Once all aspects were gathered and treated, the structure was detailed by determining its final shape and size. In terms of shape, it has been agreed, that shell like structures are superior in terms of stiffness for a given mass and that they are not to be found via topology optimization of industrial scale as discussed at length in Section “[Literature Review](#)” based on the work of Sigmund et. al. (2016).

Sizing

In this phase, the imprint of individual thicknesses on the performance of the insert was investigated. By doing so, it became obvious that the thickness distribution of the upper and lower face of the insert mainly trigger the magnitude of the resulting

Fig. 10 Sizing of insert.
Along the green arrows, the thickness decreases from two to half a millimeter



adhesive stresses. Via sizing of those faces, the stresses within the adhesive region can be dosed. An optimum was found by allowing the thickness to run out from two millimeters to a half millimeter thickness. The decrease was foreseen along the green arrows as given with Fig. 10.

Detailed Analysis

So as to gain further confidence, a final detailed numerical analysis was conducted, where especially the stresses in the adhesive region were of interest. For that sake, the discretization degree was considerably increased and second order finite elements were used.

The final analysis underpinned the findings, that a sizing of the insert upper and lower faces in such a fashion, that the thickness and thereby the stiffness runs out towards the edges, tremendously reduces adhesive stresses. In this case, the reduction in terms of stress was 35%. Figure 11 provides insight into the stress situation of the final insert design. In contrast with the milled design as given with Fig. 5 the reduction of stress by the smooth stiffness drop becomes obvious.

The derived final design has increased in weight, but is yet 10% lighter than the conventional milled design while displaying a considerable increase in structural performance. The latter is because of the higher effective stiffness, lower stress peaks in adhesive regions and increased robustness in terms of load direction.

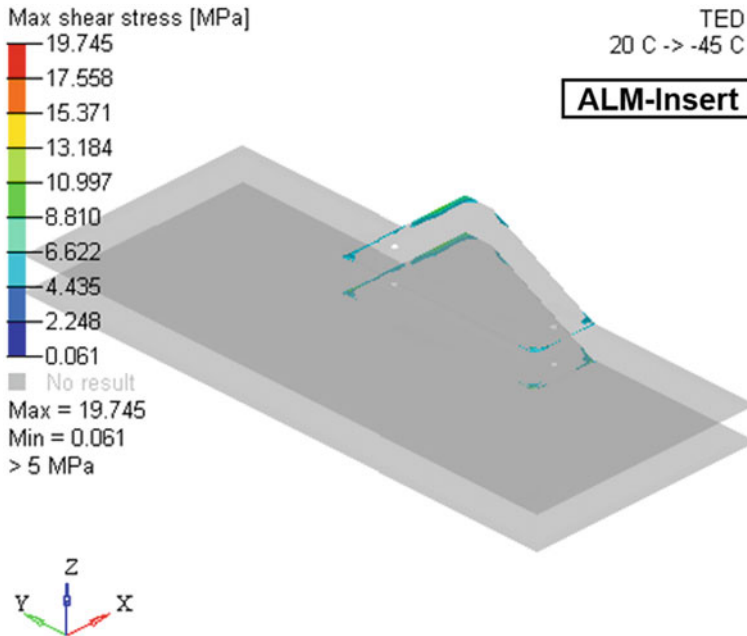


Fig. 11 Stresses in adhesive layer of 3D printed insert. The shape was derived from multiple optimizations

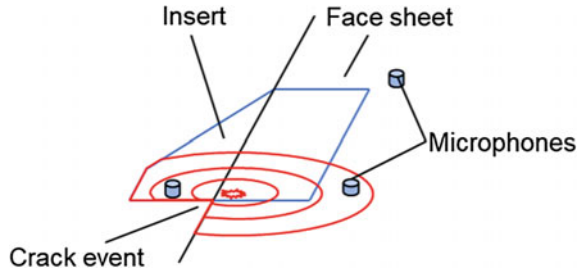
Verification Through Testing

To verify all outcomes of the analyses, a test campaign embracing all relevant flight loads, i.e. thermal and mechanical, was set-up. In addition, acoustic monitoring was used so as to properly compare the milled with the 3D printed insert.

Acoustic monitoring basically is the recording of sound waves, that originate upon any crack or damage event, by microphones. If more than two microphones are used, triangulation and simple back-calculation, then allows the determination of cracks and damages. This is schematically given with Fig. 12, wherein the upper face sheet and insert geometry are sketched as well as three microphones and a possible crack event in red. The red circles illustrate the propagation of the sound wave. By evaluating the magnitude of the acoustic emission energy a correlation to the severity of the crack event can be made.

To ensure that the whole sandwich panel is not damaged before and during the test, it was inspected by technical ultrasonography after each test run. Furthermore, two X-ray scans were made, at the beginning and at the end of the test campaign.

Fig. 12 Acoustic monitoring set-up with three microphones and a crack event in red



Mechanical Testing

The most critical mechanical test, was the application of the proof load of 20 kN onto the inserts in planar direction. For realizing this test, the sandwich panel was screwed into the test machine by using the inserts as interfaces. Prior to that, the acoustic sensors were applied so as to enable live acoustic monitoring during the test. Right before testing, the perfect alignment of the threaded rods simulating the struts was checked. The overall test set-up is given with Fig. 13.

All four inserts did withstand this load without any noteworthy events. Hence, both designs sustained the design limit load.

Thermal Test in Vacuum

Next, during thermal vacuum testing (TV), the sandwich panel had to withstand multiple thermal cycles from +55 down to -20°C in vacuum; see the following Fig. 14 (red pressure, rest temperature sensors).

During the complete test, acoustic monitoring was set-up as discussed before (see Fig. 12). With Fig. 15 the actual test set-up is visualized; whereby the thick white cables are the ones transmitting the data of the acoustic microphones. Beforehand, TV testing was regarded to be the most critical one, since cooling down by 75 K brings forth stresses as the constituent materials differ in their thermal expansions. The live monitoring along the thermal cycling already confirmed this. Final proof was established after evaluating the acoustic emission energies and number of events. The latter is simply counting noises recorded over a certain threshold. The acoustic emission energy helps to understand the severity of a single noise event. Knowing the material and how acoustic waves propagate through that material, gives a measure correlating magnitude and damage. Thus, in general very high acoustic emission energies are linked to bursts of fibers and low ones to crack initializations in the resin (inter- and intra-laminar cracks) as well as crack events in adhesive regions.



Fig. 13 Panel insert breadboard in tensile test machine

Figures 16 and 17 depict the outcome of the post-processing of the measurement data. Colors in Fig. 16, reflect the severity of the event (emission energy), where the bright green rectangles, however, show the microphones position.

The subsequent figure depicts the number of events for both inserts: 3D printed (ALM) and milled (MIL). As can be seen, the ALM part has seen way fewer events. Moreover, the average magnitude appears to be lower.

Since the load carrying capability had to be proven, the sandwich panel was mechanically loaded before and after the TV test. The change in stiffness was less than one percent and therewith structural integrity was successfully maintained.

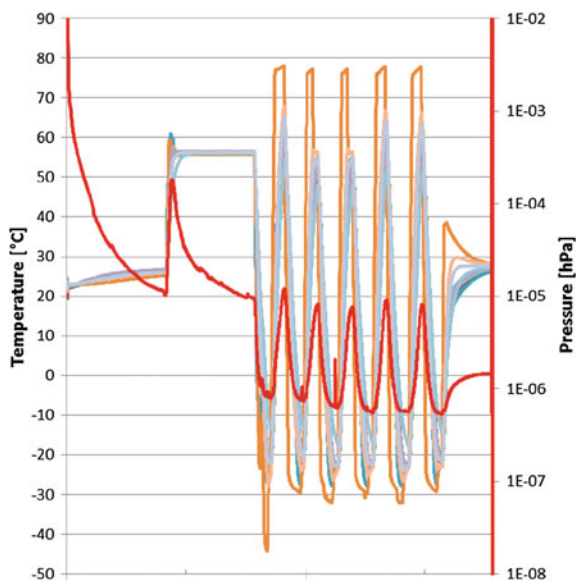


Fig. 14 Recorded temperature and pressure data of thermal vacuum test

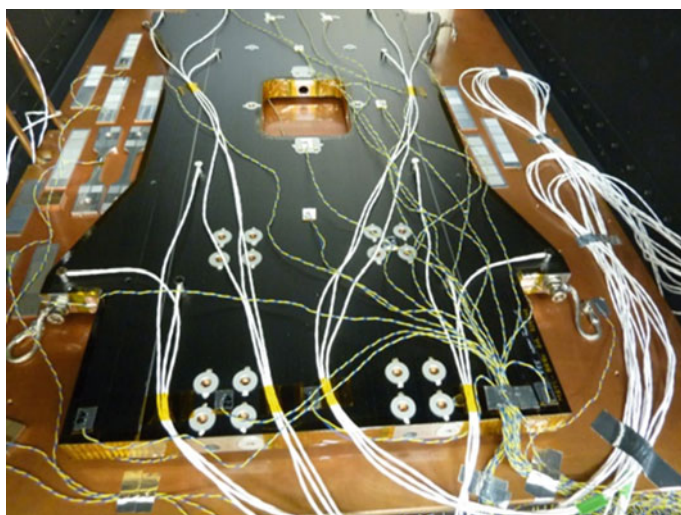


Fig. 15 Panel insert bread board in TV chamber for thermal vacuum test

Fig. 16 Post-processed records of the acoustic monitoring in terms of acoustic emission energy. Again, the magnitude as illustrated here correlates with the severity of the damage event. The light green rectangles highlight microphone positions

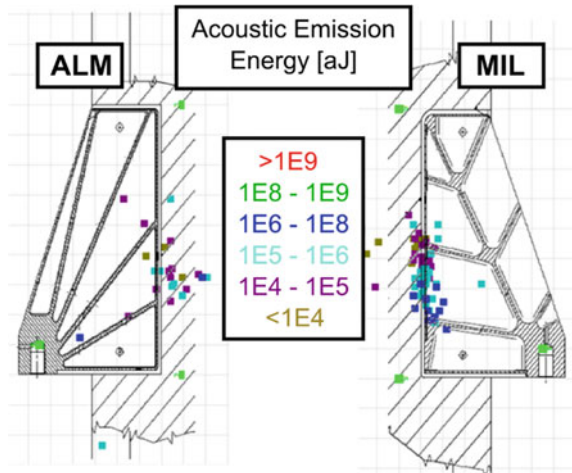
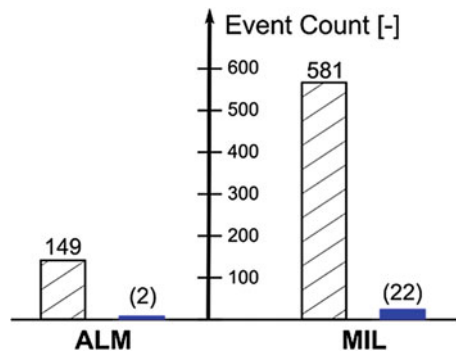


Fig. 17 Comparison of both insert designs - conventional and 3D printed - regarding number of crack events



Conclusion

As outlined, by taking advantage of topology optimization and especially its interpretation, a sustainable reduction of 10% in mass was realized. Even more relevant in the context of this work, were the stress peaks in the adhesive layer on the top and bottom face of the inserts. Those stresses were reduced by more than 30%. Extensive numerical analyses, including topology optimization runs, gave insight on how to distribute material most effectively and how to dose those bonding stresses, thereby fully exploiting 3D printing. Ultimately, a test campaign did actually prove these numerical predictions. The onset of cracks and crack events in dependency of the applied load were revealed by taking advantage of acoustic monitoring during testing. Microphones of that acoustic monitoring recorded less crack events with a lower magnitude in average. This again underpinned, that the 3D printed part is to be regarded superior in terms of load reserves owed to smaller bonding stresses.

Next, the authors focus on the qualification of 3D printing such that parts will fly in near future. In order to achieve that, more investigations and tests are necessary to show how those parts respond on cyclic loading. Topics such as crack initialization and growth (fatigue) have to be addressed as well.

Acknowledgements The METimage work described in this paper was performed as part of the industrial team led by Airbus DS GmbH on behalf of the German Space Administration DLR with funds from the German Federal Ministry of Transport and Digital Infrastructure and co-funded by EUMETSAT under DLR Contract No. 50EW1521.

References

- Bendsøe M (1989) Optimal shape design as a material distribution problem. *Struct Optim* 1:193–202
- Eschenauer H, Niels O (2001) Topology optimization of continuum structures: a review. *Appl Mech* 54:331–389
- Michell AG (1904) The limits of economy of materials in frame structures. *Philos Mag Ser 6*:589–597
- Prager W (1974) A note on discretized Michell structures. *Comput Methods Appl Mech Eng* 3:349–355
- Rozvany G, Prager W (1976) Optimal design of partially discretized grillages. *J Mech Phys Solids* 24:125–136
- Sigmund O, Aage N, Andreassen E (2016) On the (non-)optimality of Michell structures. *Struct Multi Optim* 54:361–373
- Wallner O, Reinert T, Straif C (2016) METIMAGE - a spectro-radiometer for the VII mission on board METOP-SG. In: International conference on space optics

Interactive Optimization of Path Planning for a Robot Enabled by Virtual Commissioning



Ruth Fleisch, Doris Entner, Thorsten Prante and Reinhard Pfefferkorn

Abstract Optimized path planning contributes to reducing the non-productive time of material handling in fully automated manufacturing. This paper presents a case study from the machine-tool industry sector about optimization of a path planning algorithm with the goal to minimize the time a material handling gantry robot requires to follow a feedback path, i.e. feeding a just cut part again to the saw which had just cut it, in order to realize more and more complex cutting patterns. Particularities of the case study configuration led to the application of an interactive optimization approach based on the definition and manipulation of rules for smoothing of initially planned paths and the exploration of the impacts of the rules on the time the material handling robot requires for traversing these paths by means of visual examination as well as by virtual commissioning. The achieved results were deployed in plants for cutting wooden or metal panels.

R. Fleisch (✉) · D. Entner · T. Prante
Design Automation, V-Research GmbH – Industrial Research and Development,
Stadtstraße 33, 6850 Dornbirn, Austria
e-mail: ruth.fleisch@v-research.at

D. Entner
e-mail: doris.entner@v-research.at

T. Prante
e-mail: thorsten.prante@v-research.at

R. Pfefferkorn
Schelling Anlagenbau GmbH, Gebhard-Schwärzler-Straße 34,
6858 Schwarzach, Austria
e-mail: Reinhard.Pfefferkorn@schelling.at

Introduction

Today's markets demand for customized products to fulfill increasingly client specific requirements. For cut-to-size plants, which automatically divide panels into requested sizes, the thereto necessary dealing with small lot sizes or lot-size one products results in a higher complexity of cutting patterns to enable efficient usage of the panels and thus producing as little waste as possible. As a consequence, the complexity of cut-to-size plant layouts increases as well, e.g. caused by feedbacks in their material flow (Fleisch et al. 2013). At the same time, besides high throughput and overall performance, a main requirement of operators of cut-to-size plants is a compact and space-efficient plant layout. To successfully tackle all of these requirements and challenges a new type of plant has been designed which implements material-flow feedback with reduced space usage. In order to realize the feedback loop, a gantry robot was chosen for panel transportation.

This paper reports the method and some results of minimizing the time required by this robot to move from start to end positions while at the same time obeying space limitations. Thereto, the robot must be able to move along any path and simultaneously rotate around the vertical axis while staying within spatial boundaries. While the finding of a valid path and aspects concerning the linking of the algorithm to the control system are described in detail in Fleisch et al. (2016), this paper focuses on the optimization of the path planning algorithm.

The following section describes the optimization problem and the rationales for choosing an interactive optimization approach. Section "[Gantry Robot and Integration into Plant Control System](#)" presents the material handling robot covered in the here presented case study and the integration of its control system into the overall plant control system. Section "[Related Work](#)" outlines related work, while the path planning algorithm and aspects concerning its optimization are introduced in Section "[Path Planning and Its Optimization Hook](#)". This is followed by a section about the optimization environment and procedure as well as, relating thereto, the virtual commissioning of the robot. Section "[Conclusion](#)" concludes the paper.

Optimization Problem and Approach

In order to determine which version of the path planning algorithm performs best, the following optimization loop was performed: The path planning algorithm generates a solution (a smoothed geometric path obeying certain spatial limitations), which, for evaluation, is fed into a simulation of the motion controller of the gantry robot. This simulation performs exactly the same task as the real motion controller does and therefore enables virtual production runs of a virtual plant with the real cutting patterns (i.e. virtual commissioning) towards reliable and exact temporal evaluation of the results of the respective version of the path planning algorithm. Depending on the results of this evaluation the adaption of the path planning algorithm is performed

or not (optimization loop). Adaption of the path planning algorithm is achieved in terms of definition and reworking, respectively a set of rules, which controls the smoothing of the initial result of the path planning step, i.e. a geometric path.

In other words, the objective function of the optimization problem assigns a time value calculated by means of virtual commissioning to a set of rules for smoothing a path. The behavior of the motion controller used for temporal evaluation of the results of path planning towards their optimization is unknown, i.e. a black box, in the path planning step. Hence, the optimization of a path with respect to time doesn't take place when planning a path during the operation of the plant, but instead is optimized upfront during the design and development of the robot control system in the context of overall plant (type) design and development.

The rationales for choosing an interactive approach towards fulfilling this task are the following. First, the smoothing rules to be applied in the path planning algorithm as well as their impact on the temporal behavior of the material handling robot moving from specified start to end positions were not fully understood. Both the determination of the right set of smoothing rules for geometrical manipulation of generated paths and the analysis of their impact on the temporal behavior of the robot are rather complex and partly creative tasks. Thus, in a first step, an environment needed to be created which allows knowledgeable domain experts to experiment with and understand the relationship between smoothing rules and the to-be optimized time variable. This forms the basis for designing in a next step a set of parameterized smoothing rules which will be more easily approachable by fully automated optimization procedures.

Second, it is as important for plant manufacturers to be able to exchange components or fabricators which contribute to their plants as it is not uncommon for technical systems (i.e. these components) that a new release manifests in changed behavior. Thus, plant manufacturers ensure internal process reliability by taking precautionary measures such as decoupling optimization environments from specific technical vendors and solutions, which in turn also works towards reusability of the optimization environment.

The third rationale for choosing an interactive optimization approach over a fully automated one for the described optimization task is due to the particularities of the development of control systems based on virtual commissioning. In daily practice, it is of utmost importance for gaining valid optimization results as to component behavior to keep up with the updates of each part of the overall plant system (including even overall plant process control), their interfaces and their interaction (see Fig. 8) during the rapidly changing design and development process of a plant component such as a material handling gantry robot tailored for optimized performance in cut-to-size plants employed in customer-specific manufacturing. Complementing what was stated as to the first rationale for choosing an interactive optimization approach, this makes it even more difficult to design a rich, realistic optimization model that describes all relevant aspects of the optimization problem in advance to actually performing the optimization.

These rationales led to the development of an interactive optimization environment with, on the one hand, visual support for domain expert users to analyze the impact of

smoothing rule set manipulation by comparing input and output of the path planning step and, on the other hand, reliable and exact temporal metrics provided by virtual production runs enabled by virtual commissioning.

Gantry Robot and Integration into Plant Control System

In order to realize that parts produced by sawing a panel into pieces can be fed again to the saw for further partitioning (i.e. feedback loops), in traditional cut-to-size plants, angular transfer conveyors allowing movements of panels in straight lines and turning in right angles only as well as turning devices for rotating a panel have typically been used. In new more compact and space-efficient cut-to-size plants with feedback in the material flow panels need be moved in the form of a curve while they are rotated at the same time. Therefore, a new material handling robot has been developed for transporting panels, e.g., from a saw to a feedback conveyor and vice versa (see Fig. 1).

The gantry robot picks up a panel with the aid of a suction unit and then pulls it horizontally over a brush table from a starting to an end point within a few seconds. During this movement, the robot can simultaneously rotate around the vertical axis. To prevent collisions with other panels or plant components, spatial limitations induced by the plant itself as well as by panels moving in it have to be met which can vary for each piece of material to be transported. To this end, adaptive and intelligent path planning is required to provide a fully automated and optimized transportation system with feedbacks in the material flow.

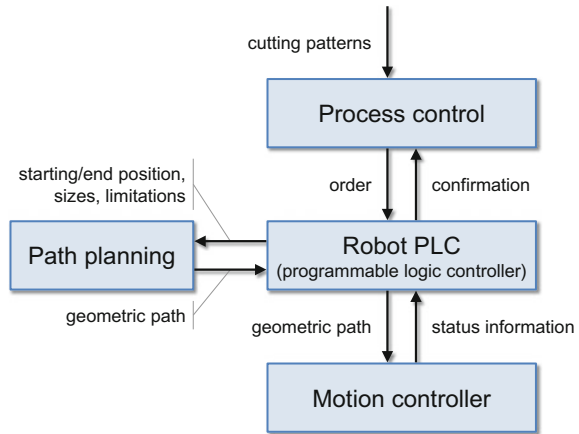
The path planning for the robot has to be incorporated into the central hierarchical control system of the plant. In the presented case study, it is linked to the robot PLC, since at this level the information necessary for the input to the path planning algorithm is available (for an overview, see Fig. 2).

The inputs to the highest layer of the control system, i.e. the process control, are cutting patterns, according to which the panels or their parts, respectively are cut by the saw and run through the plant. The process control logic controls the material flow in the plant by sending orders to PLCs (programmable logic controllers) of the different

Fig. 1 Gantry robot transporting a panel outputted by a saw towards other panels on a feedback conveyor



Fig. 2 Integration of the path planning algorithm into a hierarchical control system at PLC level



plant components like the saw itself, roller tracks, or the materials handling robot. Examples of such orders for the robot PLC are “travel to the part which is just cut and pick it up” or “transport the part from the saw to feedback conveyor and put it down”.

According to the orders, the robot PLC controls for example the suction unit of the robot or defines and transmits the input for the path planning algorithm, consisting of the starting and end position as well as panel and (robot) mechanical system size and spatial limitations. For each movement of the robot a solution path, consisting of a list of coordinates and rotation angles, has to be calculated by the path planning. This (smoothed) path is transferred to the robot PLC which passes it on to the motion controller together with other information.

The motion controller subject to this case study does not calculate a valid, collision-free geometric path itself, but requires one as input, on the basis of which it performs the trajectory planning, i.e. calculating a time-parameterized curve in a black box manner. Moreover, the motion controller controls the actuators for movements in the x-y-plane and rotations as well as sends status information back to the robot PLC.

After completing the movement of the robot from the starting to the end position, the robot PLC confirms completion of the order to the process control.

Related Work

The related work for the presented optimization of the path planning algorithm comprises three areas: motion planning, virtual commissioning as well as empirical and interactive optimization.

The problem of robot motion planning can be divided into three sub-problems: path planning, i.e. the specification of the geometric path avoiding obstacles, trajec-

tory planning, i.e. the specification of the time evolution along this geometric path, and path tracking as performed by the low-level control loops (Haschke et al. 2008). In the following, only the related work as to path planning and trajectory planning is discussed. Concerning path planning, there are several approaches like potential-field-based techniques, combinatorial methods producing roadmaps or sampling-based planning. The latter has been successfully applied in practice for many challenging problems (Siciliano and Khatib 2008) and a variant of it is realized within the project presented in this paper. Sampling-based planners rely on a collision checking module instead of explicitly representing the environment (Karaman and Frazzoli 2011) and sample the space of all possible placements of the robot for the collision-free ones (LaValle 2006).

Finding not only a feasible path for the robot, but also one that optimizes one or more criteria for a given high-level task is an important issue (Luna et al. 2013). If the optimization shall be with respect to time, trajectory planning is commonly involved. In Wu et al. (2000), for example, first, a shortest path composed of circular arcs and straight lines is obtained for a wheeled mobile robot and then, a time optimal velocity profile is generated. In Haschke et al. (2008), time-optimal trajectories are also determined based on a given geometric path. In contrast, a given geometric path is smoothed additionally in the course of planning time-optimal trajectories in Hauser and Ng-Thow-Hing (2010). In Ratliff et al. (2009), the requirement that the input path for trajectory planning has to be collision-free is even dropped. Instead, avoiding obstacles is included in planning trajectories which optimize over a variety of dynamic and task-based criteria. Finally, Shareef and Trächtler (2016) present a method for simultaneous path planning and trajectory optimization.

If it is sufficient to regard the path length as proxy for travel time (Richardson and Olson 2011), only the path planning problem itself can be considered for optimization. Besides shortening the path (Luna et al. 2013; Campana et al. 2015), different metrics like smoothness or obstacle clearance (Luna et al. 2013; Richardson and Olson 2011) or more general cost functions (Karaman and Frazzoli 2011; Campana et al. 2015) can be applied for optimizing a geometric path. Kretschmann (2007) presents an example of time-optimization, which uses geometrical features of the path, in order to avoid time-consuming calculation of the travelling time. The relationship between the geometrical parameters and the time-optimal trajectories is elaborated with the help of simulation. Theoretical considerations presented by Kretschmann prove that the cost function based on the geometrical parameters is a sufficiently reliable indicator for the travelling time.

The solution approach presented in this paper follows the idea of deciding which rules are best for smoothing the geometric paths within path planning by evaluating the rules as to temporal behavior of the robot. This optimization procedure is enabled by virtual commissioning of the robot, which involves the trajectory planning.

The purpose of virtual commissioning is to test manufacturing systems together with their control programs through simulation to produce reliable and precise behavioral forecasts before on-site installation and ramp-up (Hoffmann et al. 2010). Beyond that, virtual commissioning enables optimization in relation to control systems. In Svensson et al. (2012), for example, a simulation-based optimization method is pre-

sented where a combination of real industrial control systems and of a virtual manufacturing system is used for the simulation part. In their case study based on this test bed, optimization was applied for tuning process parameters of an automotive sheet-metal press line.

Empirical optimization is used in the field of code compiling, especially for library generation, to choose parameter values or even an algorithm from a suite of algorithms by generating different versions of the program, running all of them on the actual hardware and selecting the version which results in the best performance (Yotov 2003; Epshteyn 2006). This idea of evaluating multiple code versions is also applied in the project presented in this paper, however, a part of the test system is not the real one but virtual, i.e. the motion controller simulation.

Interactive optimization means that the user participates actively in the optimization process. More precisely, “an interactive approach recognizes some limits to modeling and parameter setting in a real situation, and values the user’s expertise in the application domain that can be exploited by the optimization system” (Meignan et al. 2015). This means, that “with an adequate interaction between an optimization system and its users, the optimization model can be enriched to fit the real problem, the search process can be guided for improving its efficiency, and the user can better understand the system” (Meignan et al. 2015).

In Meignan et al. (2015), a classification of interactive optimization methods is proposed according to the purpose of the interaction and the role of the user: In the case of problem-oriented interaction, the user aims at modifying the optimization problem by adjusting the existing constraints or objectives or by enriching the problem, i.e. defining new constraints or objectives. The target of search-oriented interaction is improving the performance of the optimization procedure. The user can achieve this by tuning strategic parameters of the optimization procedure, by supporting the search with information related to decision variables (guiding), or by acting as a search procedure (assisting).

The here presented optimization approach belongs to the assisting-category, as the user has to define, implement and rework a set of smoothing rules, that is the next element of the solution space for which the time has to be calculated. On the other hand, the approach belongs to the guiding-category as well, because the user analyzes the link between the implemented set of rules and the time for executing the path and uses this information to change the set of rules.

Path Planning and Its Optimization Hook

Before discussing the set of smoothing rules as the hook for optimizing travel time of the material handling robot for a path computed by the path planning algorithm, this section, first, summarizes details of the path planning algorithm and its specific interplay with the motion controller deployed in this case study. Further information on the latter topic can be found in Fleisch et al. (2016).

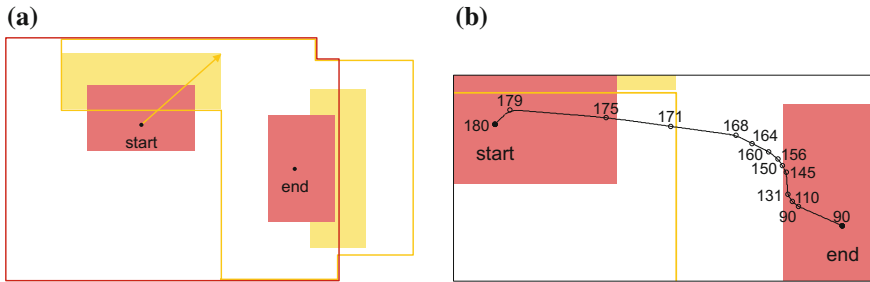


Fig. 3 **a** Panel (yellow, size: 2160 mm \times 750 mm, arrow marks the offset) and part of the mechanical system (red), their limitations (solid lines), and rotation point in starting and end position. **b** A solution path for the rotation point of the robot. The rotation angles for each point are given in degrees

Path Planning and Its Interplay with the Motion Controller

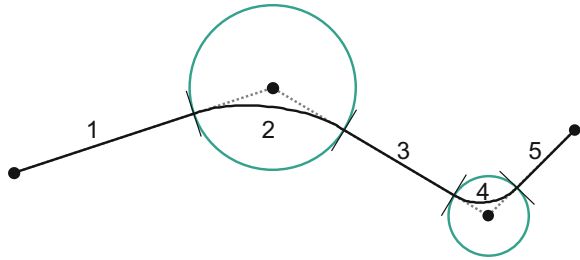
The developed path planning algorithm computes a smoothed geometric path for every movement of the material handling robot from a starting to an end point in the horizontal plane. Such a path avoids obstacles and contains information for simultaneous rotation around the vertical axis of the robot. The movement can take place with or without material, the latter, e.g., in order to collect a panel.

The path planning algorithm comprises the following three steps: (step 1) computation of a valid geometric path, (step 2) smoothing this path (see Fig. 6), and (step 3) determining a feasible blending parameter for each position of the path (see Fig. 4). Subsequently, the final curve to be travelled along by the robot (i.e. trajectory planning) is determined by its motion controller, which calculates in a black box manner a time-parameterized curve corresponding to the input it received from the path planning algorithm.

Here, we first describe input and output of the path planning algorithm, followed by the challenges related to the black box calculation of the time-parameterized curve by the motion controller and how they influence the path planning algorithm.

Figure 3a visualizes the main input data to the path planning algorithm of a real-world example: A panel to be transported is always rectangular and must not exit a given area. This ensures that collisions are prevented, e.g. of the panel with other panels in the plant or with plant components. A part of the mechanical system of the robot is also represented as a rectangle and must stay inside its own limitations. The admissible areas for the rectangles are simple polygons, featuring internal angles of 90° or 270° . Thus, in the case of transporting a panel, the main input parameters for the path planning algorithm are the sizes of both rectangles, the offsets of the rectangles relative to the rotation point of the robot, the spatial limitations, and the starting and end position. A position is composed of the x- and y-coordinates of the rotation point of the robot in the horizontal plane and further includes the associated rotation angle for this point.

Fig. 4 The respective blending parameters are the radii of the depicted circles. 1, 3 and 5 are linear segments, 2 and 4 are polynomial segments



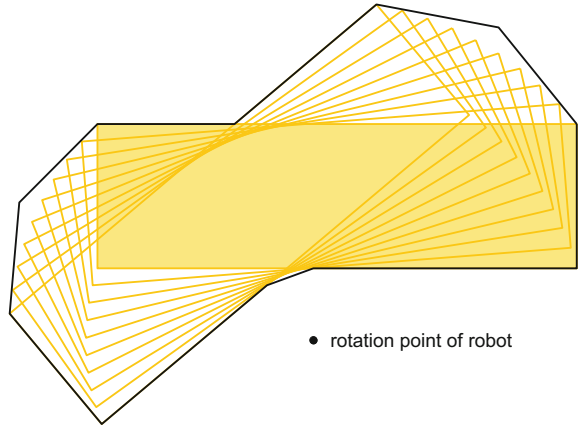
The output of the path planning algorithm is a smoothed geometric path for the rotation point of the robot from the starting to the end position such that the relevant spatial limitations are not violated (see Fig. 3b). The geometric path is a list of positions where the number of positions must not exceed 25—this requirement is introduced by the motion controller. Each position is enriched by a blending parameter (step 3) which indicates the extent of blending of the movement along the polygonal chain at the respective point (see Fig. 4).

The list of positions together with the blending parameter for every point is passed onto the motion controller of the robot, where it serves as a basis for generating a trajectory and for controlling the movement. The points of the path in the x-y-plane are connected by linear interpolation with a polynomial blend (see Fig. 3b). Thus, the plane curve consists of linear and polynomial segments.

If the polygonal chain is blended at a point, the rotation angles corresponding to the transition points between the blending segment (e.g. segment 2 in Fig. 4) and the preceding (segment 1) as well as the subsequent (segment 3) linear segment are calculated by comparing the proportions of the path length. However, the rules of how the rotation angle is interpolated are not available for path planning, since they are implemented in conjunction with the trajectory planning in the motion controller. Thus, the matching of rotation angles to the plane curve can be computed only at the transition points of the segments. Therefore, it has to be assumed that, in theory, the rotation to be executed along a segment can take place at any arbitrary point of that segment. This has an effect as to collision avoidance and can result in, e.g., a higher number of positions required for specifying a path in case of tight spatial limitations and only little space left to route. Dealing with this effect is explained below.

For computing a valid geometric path a variant of a sampling-based planner was implemented which samples the space of all possible placements of the robot for the collision-free ones (Fleisch et al. 2016). The sampling is realized as an incremental search, where the global search direction is along the medial axis of the polygonal limitation. In the case that a panel is moved, the admissible area of the panel is used for determining the medial axis as the material is usually more critical than the mechanical system regarding the available space. In a step of the incremental search, the next feasible position is found by varying the position or the step size. For the purpose of checking the feasibility of the next position, i.e. collision detection, the area which is covered by a rectangle rotating from the starting to the end angle of

Fig. 5 Polygon (black) encompassing a rotating rectangle



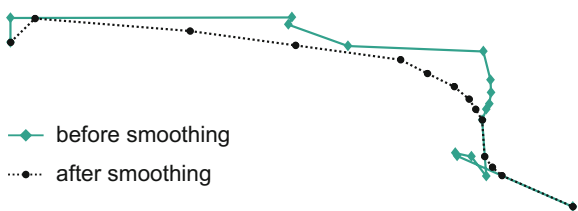
the step is approximated by a surrounding polygon (see Fig. 5). This polygon must not be outside the limitations when it is moved along the segment from the current to the next point.

Optimization Hook: Set of Smoothing Rules

As the behavior of the motion controller concerning trajectory planning and interpolation of the rotation angle is unknown, i.e. a black box, in path planning (steps 1-3), the optimization of a path with respect to time is not possible within the path planning step. Instead, improvements can only be based on geometrical features of the path. Since a smoother path is assumed to be typically more time-efficient, the task to be tackled by interactive optimization is to determine which rules for smoothing the paths shall be implemented in order to minimize the time which the robot needs for following the paths.

As elaborated in Section “[Path Planning and Its Interplay with the Motion Controller](#)”, the path planning algorithm integrates the smoothing by first computing a valid geometric path (step 1), then smoothing the solution (step 2) and, finally, determining a feasible blending parameter for each position (step 3). Figure 6 shows an example of an initial valid path and, based on it, a smoothed path.

Fig. 6 Valid path before and after smoothing (rotation angles are not depicted)



The input for the smoothing part of the algorithm is a valid geometric path, which means a polygonal chain together with a rotation angle for each of its points. Following this path ensures that both the panel and the modeled part of the mechanical system do not exit their admissible areas. The polygonal chain is iteratively smoothed by eliminating or modifying positions based on smoothing rules while maintaining the spatial limitations. An example for a rule which chooses a position for eliminating or changing, is the selection of the most acute angle, formed by the line segments of the polygonal chain. Further options for changing a position are for instance translating a point to the centroid of the triangle with the chosen point and the preceding as well as the subsequent point of the polygonal chain as vertices or replacing a position by the center of the preceding and subsequent position (both the center in the x - y -plane and the center of the rotation angle). Further rules are based among other aspects on the following heuristics for smoothing a path:

- the elimination of self-intersections of the polygonal chain,
- shortening the total length of the polygonal chain,
- avoidance of short line segments of the polygonal chain,
- the reduction of the number of positions, or
- a more uniform distribution of the rotation angles along the polygonal chain.

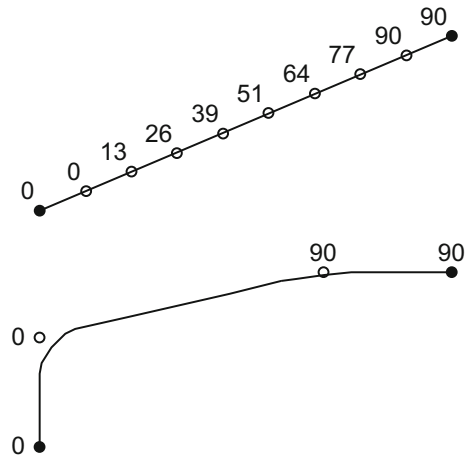
For the determination of the best rules to be implemented, an interactive optimizing procedure is applied which includes virtual production runs.

Optimization Environment and Procedure

As illustrated just before, the optimization as to the time the material handling gantry robot needs to follow a path is carried out by interactive definition of the smoothing rules. One of the challenges in doing so comes from the fact that it is not necessarily the shortest path yielding the shortest travel time since, for example, the robot has to slow down for turning sharply in the x - y -plane and hence a longer path with obtuse angles can be preferable to a shorter one with acute angles. Thus, the optimization of the path planning provides a possibility to find a trade-off between the path length and the sizes of the path angles for example.

Another illustrative example which seems counter intuitive at first sight is depicted in Fig. 7. The two shown paths are the outcome for the same input data but determined by two different sets of smoothing rules. This example again reveals that a mere visual examination does not suffice but the exact calculation of times by virtual commissioning of the robot is relevant, since the latter delivers an unexpected result: Due to the behavior of the motion controller, a straight line in the x - y -plane defined by 10 positions is inferior regarding time to a non-linear, longer path specified by only 4 positions, where both paths represent a rotation by 90° and the distance between the starting and end point is 1.69 m. It takes the robot 8.04 s to follow the straight line and 7.21 s to follow the non-linear path, yielding a difference of about 10%. This illustrates that the number of positions for defining a path has a great impact on the

Fig. 7 It takes the robot more time to follow the first path than to follow the second one. Rotation angles in degrees are depicted



material-handling time for the robot (motion controller) at hand. The high number of positions of the linear path is required for distributing the rotation angle along the straight line due to the close proximity of the boundaries.

To get a first glance of the impact of the smoothing rules on the geometry of the paths, the paths are visually examined. In addition, for the optimization, the exact execution times for moving the panels are determined. Since this calculation and data concerning temporal aspects such as velocity, acceleration, or jerk are not available at the path planning level, the motion controller of the material handling gantry robot with its trajectory planning is necessary, as the times depend on its behavior. Consequently, the evaluation with respect to time of the implemented rules is performed by virtual commissioning of the robot.

Virtual Commissioning of the Material Handling Robot

Apart from selecting the best rules for smoothing within the frame of path planning and thereby optimizing the material handling robot with regard to its temporal behavior, virtual commissioning of the robot facilitates also the general development of the path planning algorithm, testing its integration into the control system as well as the coordination of the interaction between the robot PLC and the motion controller. This leads to improved quality of the algorithm and reduced commissioning time and costs of the plant.

In order to establish a system for optimizing the path planning, an industrial computer has been set up where the process control, the PLC and the path planning algorithm are installed. The same computer can be used later in the real plant as well. In contrast, a simulator of the motion controller is employed in place of the real one. Figure 8 depicts the components of the system for virtual commissioning of the robot.

Fig. 8 For optimization, a simulator of the motion controller is used

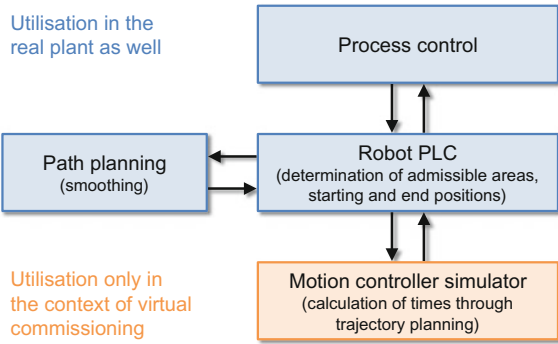
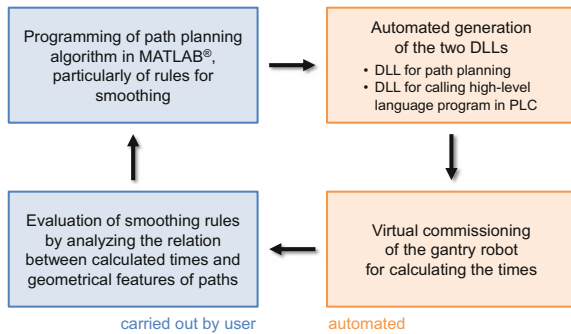


Fig. 9 Optimization loop



The challenge related to virtual commissioning of the robot is to keep up with the updates of each component of the system, their interfaces and their interaction during the rapidly changing design and development phase of the robot.

Details of the Implementation and of the Optimization Loop

As evaluating of the path planning algorithm in the frame of interactive optimization should be as efficient as possible, the path planning algorithm is implemented by using a high-level programming language instead of directly programming in PLC code. This is more suitable for developing complex algorithms. In the present case, the algorithm is developed in MATLAB®. MATLAB Coder™ generates C and C++ code from MATLAB® code and, in a second step, optionally machine-readable code in the form of a DLL (dynamic link library). In order to call high-level language programs from the control program, another DLL is required. The generation of the two DLLs based on the high-level programming language code of the algorithm is automated so as to enable more efficient working with the optimization system and that modifications of the rules in the algorithm are ready for evaluation quickly. Figure 9 gives an overview of the interactive optimization procedure.

The loop starts with defining the smoothing rules and programming them in MATLAB[®]. This has to be accomplished by the user, whereas the two DLLs are automatically generated on the basis of the MATLAB[®] code and are made available for deployment on the system for virtual commissioning. In order to obtain the traversal times of the robot, a production run is simulated by means of the virtual commissioning system. For this, the production list in form of cutting patterns is loaded in the graphical user interface of the process control and the virtual production is started. A log file is created for every path which contains the input and output data of the path planning algorithm and data recorded by the simulator of the motion controller. The recorded data is a list of time stamps and for each time stamp the corresponding position of the robot (x- and y-coordinates of the rotation point and rotation angle). With this data, the movement of the robot (with or without a panel) following the path can be visualized. It is the same motion as in the real world or as depicted in the graphical user interface of the process control, but with the advantage that the spatial limitations and the solution path are illustrated. The visualization supports the analysis of the relation between the calculated times and the geometrical features of the path and thus, facilitates definition and evaluation of the smoothing rules. Based on the findings, the set of smoothing rules can be reworked.

The MATLAB[®] code of the algorithm is structured in such a way that only local alterations are necessary for implementing new smoothing rules which keeps the effort to a minimum and contributes to the robustness of the code. Together with the automated generation of the DLLs, this achieves reduced amount of user input for optimization and thus increases the efficiency of the optimization loop.

Conclusion

The trend towards customer-specific production, for example in the furniture industry, has led to the development of cut-to-size plants with feedbacks in their material flow. In order to realize feedbacks in a compact and space-efficient way as well as to achieve the target of time-optimized panel handling, a new material handling gantry robot has been developed. This paper presented an interactive approach to optimizing the path planning algorithm of the robot as to its temporal behavior when traversing planned paths. Thereto, the smoothing rules applied to initially generated paths were used as optimization hook. As traversal times of the robot are not available in the path planning step, a system for virtual commissioning of the robot was established. It enables virtual production runs and thereby the temporal evaluation of the smoothing rules in order to find the optimal ones as to their impacts on the behavior of the robot. During the optimization loop user input is required, on the one hand, for analyzing the relation between geometrical features of a path and the calculated time which the robot needs to follow the path and, on the other hand, for adapting the smoothing rules of the path planning algorithm. Virtual commissioning has the advantage that it can be conducted already during the design and development process of the materials handling gantry robot and not as late as during commissioning on a customer's site.

Potential extensions and future work concerning path planning optimization include the automation of the evaluation of the smoothing rules (lower box on the left-hand side of Fig. 9) to improve and accelerate the utilization and operation of the optimization system.

Acknowledgements This work was carried out within the COMET K-Project #843551 “Advanced Engineering Design Automation (AEDA)” funded by the Austrian Research Promotion Agency FFG.

References

- Campana M, Lamiroux F, Laumond J-P (2015) A simple path optimization method for motion planning. Rapport LAAS n 15108
- Epshteyn A et al (2006) Analytic models and empirical search: a hybrid approach to code optimization. In: Proceedings of the 18th international conference on languages and compilers for parallel computing. Springer, Berlin, pp 259–273
- Fleisch R, Schöch R, Prante T, Pfelegerl R (2013) Consistent use of emulation across different stages of plant development—The case of deadlock avoidance for cyclic cut-to-size processes. In: Winter simulation conference (WSC 2013), pp 2565–2576
- Fleisch R, Schöch R, Prante T, Pfefferkorn R (2016) A path planning algorithm for a materials handling gantry robot and its validation by virtual commissioning. In: Advances in manufacturing technology XXX, pp 169–174
- Haschke R, Weitnauer E, Ritter H (2008) On-line planning of time-optimal, jerk-limited trajectories. In: 2008 IEEE/RSJ international conference on intelligent robots and systems, pp 3248–3253
- Hauser K, Ng-Thow-Hing V (2010) Fast smoothing of manipulator trajectories using optimal bounded-acceleration shortcuts. In: 2010 IEEE international conference on robotics and automation, pp 2493–2498
- Hoffmann P, Schumann R, Maksoud TM, Premier GC (2010) Virtual commissioning of manufacturing systems a review and new approaches for simplification. In: ECMS, pp 175–181
- Karaman S, Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. *Int J Rob Res* 30:846–894
- Kretschmann R (2007) Zeitoptimale Bahnplanung für Industrieroboter. Available at: [http://www.mftech.de/diplom_zeitoptimale_bahnplanung_industrieroboter_uebersicht\(2007\).pdf](http://www.mftech.de/diplom_zeitoptimale_bahnplanung_industrieroboter_uebersicht(2007).pdf). Accessed 27 Apr 2017
- LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge
- Luna R, Şucan IA, Moll M, Kavraki LE (2013) Anytime solution optimization for sampling-based motion planning. In: 2013 IEEE international conference on robotics and automation, pp 5068–5074
- Meignan D, Knust S, Frayret J-M, Pesant G, Gaud N (2015) A review and taxonomy of interactive optimization methods in operations research. *ACM Trans Interact Intell Syst* 5:17:1–17:43
- Ratliff N, Zucker M, Bagnell JA, Srinivasa S (2009) CHOMP: gradient optimization techniques for efficient motion planning. In: IEEE international conference on robotics and automation
- Richardson A, Olson E (2011) Iterative path optimization for practical robot planning. In: 2011 IEEE/RSJ international conference on intelligent robots and systems, pp 3881–3886
- Shareef Z, Trächtler A (2016) Simultaneous path planning and trajectory optimization for robotic manipulators using discrete mechanics and optimal control. *Robotica* 34:1322–1334
- Siciliano B, Khatib O (2008) Springer handbook of robotics. Springer Science & Business Media, Berlin
- Svensson B, Danielsson F, Lennartson B (2012) Time-synchronised hardware-in-the-loop simulation—Applied to sheet-metal press optimisation. *Control Eng Pract* 20:792–804

- Wu W, Chen H, Woo P-Y (2000) Time optimal path planning for a wheeled mobile robot. *J Rob Syst* 17:585–591
- Yotov K et al (2003) A comparison of empirical and model-driven optimization. In: *Proceedings of the ACM SIGPLAN 2003 conference on programming language design and implementation*. ACM, New York, pp 63–76

Box-Type Boom Design Using Surrogate Modeling: Introducing an Industrial Optimization Benchmark



Philipp Fleck, Doris Entner, Clemens Münzer, Michael Kommenda, Thorsten Prante, Martin Schwarz, Martin Hächl and Michael Affenzeller

Abstract Simulation-based optimization problems are often an inherent part in engineering design tasks. This paper introduces one such use case, the design of a box-type boom of a crane, which requires a time consuming structural analysis for validation. To overcome high runtimes for optimization approaches with numerous calls to the structural analysis tool, we here present several ways of approximating the structural analysis results using surrogate models. Results show a strong correlation between certain statics input and output parameters, and that various surrogate modeling approaches yield similar results in terms of accuracy and impact of the

P. Fleck (✉) · M. Kommenda · M. Affenzeller
Heuristic and Evolutionary Algorithms Laboratory, University of Applied
Sciences Upper Austria, Softwarepark 11, 4232 Hagenberg im Mühlkreis, Austria
e-mail: philipp.fleck@fh-hagenberg.at

M. Kommenda
e-mail: michael.kommenda@fh-hagenberg.at

M. Affenzeller
e-mail: michael.affenzeller@fh-hagenberg.at

D. Entner · T. Prante
Design Automation, V-Research GmbH, CAMPUS V,
Stadtstraße 33, 6850 Dornbirn, Austria
e-mail: doris.entner@v-research.at

T. Prante
e-mail: thorsten.prante@v-research.at

C. Münzer
Engineering Design and Computing Laboratory, ETH Zurich,
Tannenstrasse 3, 8092 Zurich, Switzerland
e-mail: muenzerc@alumni.ethz.ch

M. Schwarz · M. Hächl
Liebherr-Werk Nenzing GmbH,
Dr.-Hans-Liebherr-Straße 1, 6710 Nenzing, Austria
e-mail: martin.schwarz@liebherr.com

M. Hächl
e-mail: martin.haechl@liebherr.com

predictors on the output. The box-type boom use case together with the surrogate models shall serve as an industrial optimization benchmark for comparing various algorithms on this simulation-based optimization problem.

Introduction

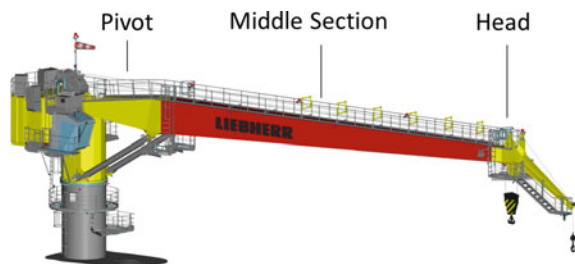
In competitive markets, companies often have to offer customer-specific products to meet clients' requirements. This frequently involves costly new or re-design of existing products. One possibility to keep the costs at an affordable level is *Engineering Design Automation* (EDA). According to Hubka and Eder (1987), "Engineering design is a process performed by humans aided by technical means through which information in the form of requirements is converted into information in the form of descriptions of technical systems, such that this technical system meets the requirements of mankind". This process is (partly) automated in EDA.

Simulation and optimization tasks are often an inherent part of engineering design problems, since the ways of defining and evaluating a product often become too complex to find valid and (near) optimal solutions manually in reasonable time (Deb 2010; Roy et al. 2008). A wide range of industries have adopted optimization methods within EDA, e.g. for validating and improving structural aspects (Affenzeller et al. 2015), or appropriately selecting, dimensioning and assembling components to fulfill customer-specific needs (Zhang 2014).

The focus of this paper is a case study concerning the engineering design of a Box-Type Boom (BTB) crane, commonly seen in maritime applications (see Fig. 1 for an example). Such cranes are a prime example for requiring re-design for almost every ordered boom due to varying customer requirements such as boom length and load cases.

Together with Liebherr-Werk Nenzing (2017), the automation of the BTB design in terms of automatically generating a 3D-CAD model, production drawings and welding plans of the boom based on case-specific input values (e.g. length, statics parameters) was realized earlier (Frank et al. 2014). A major task left to the engineer is the definition of the statics parameters (e.g. thicknesses of the plates), which are chosen manually based on experience and previously designed booms. Each boom design must be evaluated using a structural analysis tool to verify static requirements (e.g. stress).

Fig. 1 Box-type boom crane with indicated pivot- and head-part as well as middle section



The contribution of this paper is two-fold: First, the BTB optimization problem for automating and optimizing the manual process of defining the statics parameters while minimizing costs is presented in Section “[Box-Type Boom Optimization Problem](#)”. The case study shall later serve as an industrial benchmark for comparing different optimization approaches in a simulation-based optimization environment. In comparison to other popular design optimization benchmark problems, such as the pressure vessel problem (Sandgren 1988) or the welded beam problem (Ragsdell and Phillips 1976), the BTB optimization problem is based on a real-world industrial use case that requires a runtime expensive simulation for evaluating certain constraints.

The second contribution of this paper deals with the issue of this runtime expensive simulation tool for the structural analysis of the BTB (presented in Section “[Surrogate Modeling for the Box-Type Boom](#)”). Since optimization approaches potentially make numerous calls to this structural analysis tool, surrogate models (Forrester and Keane 2009; Wang and Shan 2007) are learned to replace the runtime expensive evaluation in an optimization procedure. Additionally, relationships between the input (e.g. thicknesses of plates) and output (e.g. utilization with regard to stress) of the structural analysis tool can be investigated by analyzing the surrogate models (presented in Section “[Results](#)”).

Overall, the paper comprises to the introduction of the BTB optimization benchmark problem and the surrogate modeling for replacing the structural analysis tool. Automation and optimization based on these results is out of scope of the paper and left for future work.

Box-Type Boom Optimization Problem

A BTB consists of a pivot-, a middle and a head-section (see Fig. 1). The pivot- and head-part are selected from a list of standardized parts; thus, the optimization of the BTB is limited to the middle section of the boom. This middle section is further divided into segments (3m each), each of which lies between two bulkheads or a bulkhead and the pivot- or head-part (see Fig. 2 for a boom with 5 segments).

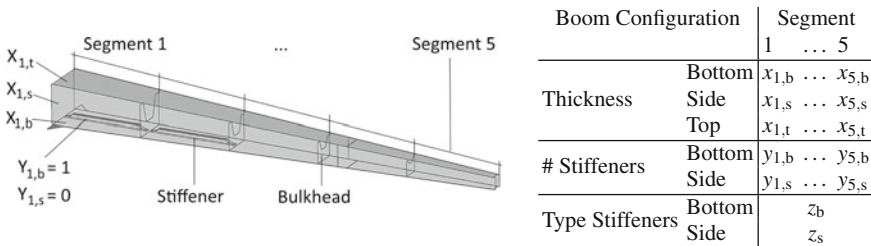


Fig. 2 Box-type boom optimization: variables of a boom configuration

Each segment is described by five variables: Thickness of the bottom, side (same for both sides) and top plates as well as the number of stiffeners on the bottom and side plates. Furthermore, there are two global variables, the type of stiffeners on the bottom plates and on the side plates. One setting of these variables makes up a *boom configuration* (see Fig. 2). Formally, with n being the number of segments, we define $x_{i,j}$, $y_{i,k}$, z_k with $i \in \{1, \dots, n\}$, $j \in \{b, s, t\}$, $k \in \{b, s\}$ as follows:

- $x_{i,j}$ – plate thickness of segment i on the bottom ($j = b$), side ($j = s$) or top ($j = t$)
- $y_{i,k}$ – number of stiffeners of segment i on the bottom ($k = b$) or side ($k = s$)
- z_k – type of stiffeners on the bottom ($k = b$) or side ($k = s$)

The objective of the optimization is to find a boom configuration with minimal material and welding costs while satisfying a set of constraints (see below). The material costs include the material of the plates and the stiffeners. The welding costs result from welding the stiffeners to the plates as well as welding the plates to form the boom. For the latter part, the plates of the segments, which are of length 3 meters or less (for the last segment), can be combined to larger plates of up to 10 meters. Thus, the welding costs are approximated by using the length of the weld seam and the thicknesses of the combined plates. For $\mathbf{x} = (x_{1,b}, x_{1,s}, x_{1,t}, x_{2,b}, x_{2,s}, x_{2,t}, \dots, x_{n,b}, x_{n,s}, x_{n,t})$, $\mathbf{y} = (y_{1,b}, y_{1,s}, y_{2,b}, y_{2,s}, \dots, y_{n,b}, y_{n,s})$ and $\mathbf{z} = (z_b, z_s)$, the objective function is defined as

$$f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{i=1}^n \left(\sum_{j \in \{b,s,t\}} mc_{plate}(x_{i,j}) + \sum_{k \in \{b,s\}} y_{i,k} mc_{stiffener}(z_k) + \sum_{k \in \{b,s\}} y_{i,k} wc_{stiffener}(z_k) \right) + wc_{boom}(\mathbf{x}), \quad (1)$$

with

- mc_{plate} material costs of the plate depending on the thickness $x_{i,j}$ of the i^{th} segment on the bottom, side or top, respectively;
- $mc_{stiffener}$ material costs of the stiffener depending on the stiffener type z_k on the bottom or side, respectively;
- $wc_{stiffener}$ welding costs of the stiffener depending on the stiffener type z_k on the bottom or side, respectively; and
- wc_{boom} welding costs of the plates depending on all thicknesses $x_{i,j}$, $i \in \{1, \dots, n\}$, $j \in \{b, s, t\}$.

In order to design a functioning crane, the boom configuration has to be chosen such that certain statics constraints as well as constraints on the plate thicknesses, and number and type of stiffeners are fulfilled. These constraints are formally defined within the optimization problem below:

minimize $f(\mathbf{x}, \mathbf{y}, \mathbf{z})$
 $x_{i,j}, y_{i,k}, z_k$

subject to

$$\begin{aligned}
 \sigma_{i,j} < 1 & \quad (2) & x_{i,b} \in \{\rho_0, \dots, \rho_p\} \subset \mathbb{N} & \quad (5) & y_{i,b} \in \{0, 1, 2, 3\} & \quad (9) \\
 \phi_{i,j} < 1 & \quad (3) & x_{i,s} \in \{\tau_0, \dots, \tau_q\} \subset \mathbb{N} & \quad (6) & y_{i,s} \in \{0, 1, 2\} & \quad (10) \\
 \beta_{i,k} < 1 & \quad (4) & x_{i,t} \in \{\omega_0, \dots, \omega_r\} \subset \mathbb{N} & \quad (7) & y_{i,k} \geq y_{i+1,k} & \quad (11) \\
 & & x_{i,j} \geq x_{i+1,j} & \quad (8) & z_k \in \{1, 2, 3, 4\} & \quad (12)
 \end{aligned}$$

for all $i \in \{1, \dots, n\}, j \in \{b, s, t\}, k \in \{b, s\}$

The objective function, defined in Eq. (1), shall be minimized subject to a set of constraints. The constraints in Eqs. (2) to (4) state that the utility constraints with respect to stress (denoted by σ) and fatigue (ϕ) on the bottom, side and top as well as with respect to buckling (β) on the bottom and side are fulfilled for each segment. A value larger than 1 is a violation, and the larger the value, the higher the violation. A structural analysis tool is used to calculate these utilizations for a given boom configuration for predefined load-cases and constraints-margins according to the customers' needs. In Eqs. (5) to (7) the allowed thicknesses for the plates on the bottom, side and top, respectively, are defined as a subset of the natural numbers. For instance, for the bottom, the thicknesses could be defined as a set of 5 values $\{\rho_0, \rho_1, \rho_2, \rho_3, \rho_4\}$. Equation (8) states that these thicknesses are non-increasing from the pivot- to the head-part. The number of stiffeners is limited by the constraints in Eqs. (9) and (10) to lie between 0 and 3 (bottom), and 0 and 2 (side), and is non-increasing from the pivot- to the head-part (Eq. (11)). Finally, the four allowed types of stiffeners (U-shaped with different dimensions) are stated in Eq. (12). For the remainder of the paper, all constraints are considered hard constraints.

Surrogate Modeling for the Box-Type Boom

Solving optimization problems with runtime expensive solution evaluations usually results in total execution times that are too high for practical applications. Considering the example of a 9-segment boom, one boom configuration consists of $5 \cdot 9 + 2 = 47$ variables. Assuming five possible thicknesses for each bottom, side and top (\mathbf{x}), there are $5 \cdot 3 \cdot 9$ possibilities of assigning these thicknesses (disregarding any other constraints). Furthermore, there are $4 \cdot 9 = 36$ possibilities for stiffener arrangements on the bottom and $3 \cdot 9 = 27$ on the side (\mathbf{y}), and $2 \cdot 4 = 8$ choices of stiffener types (\mathbf{z}) (again disregarding any other constraints). Thus, there are in total more than a million solution candidates. Evaluating only 10,000 of these candidates with the structural analysis tool at hand (with a runtime of 5 min each) would already take about a month.

Surrogate modeling (Wang and Shan 2007; Forrester and Keane 2009) can be used to make common optimization techniques feasible, by replacing expensive calculations (such as the structural analysis for the BTB) with simpler models. Typically, machine learning (Bishop 2006) techniques, such as linear regression, support vector regression, neural networks, Gaussian processes and symbolic regression, are used to learn surrogate models. These methods use data of previously performed expensive evaluations to learn a model that predicts the value of unseen evaluations.

For the box-type boom, the *input variables* variables of the models are the thicknesses (\mathbf{x}), and the number and type of stiffeners (\mathbf{y} and \mathbf{z} , respectively). The *output variables* are the statics constraints (eight per segment): Degree of utilization of stress (bottom, side, top), fatigue (bottom, side, top) and buckling (bottom, side). Continuing the 9-segment boom example, there are $8 \cdot 9 = 72$ statics constraints. The surrogate models will be trained based on those input and output to predict the statics constraints of boom configurations that are not yet evaluated.

Before the surrogate models are trained, the sampled data is analyzed to gain insights in the correlations between the input and output variables (Section “[Data Analysis](#)”). Based on this analysis, appropriate subsets of the input variables are suggested, and ways of reducing the number of models which have to be trained are investigated (Section “[Variable Selection and Data Preprocessing](#)”). Finally, the modeling procedure is described (Section “[Modeling Procedure](#)”).

The data analysis and modeling was done in the open-source framework HeuristicLab (HeuristicLab 2017; Wagner et al. 2014), which provides a large set of popular heuristic and evolutionary optimization algorithms and also provides powerful tools and algorithms for data analysis and data-based modeling. The data on which the presented results are based on include a set of approx. 5000 samples, which were created with HeuristicLab’s distributed computation environment.

Data Analysis

We first investigated whether there are measurable correlations between the sampled inputs and outputs to better understand the relations between the variables of the BTB problem and to assess which variables are important for the modeling. Because the inputs are fixed integers, we used Spearman’s rank (Lehmnn et al. 2005) as correlation measure, as opposed to the popular Pearson R.

Figure 3 shows the correlations between the inputs and outputs of segments 3 and 4, along which we first investigate the correlations within a single segment. Green color indicates a positive correlation, red a negative one and white no correlation. The highest correlation can be observed between bottom fatigue $\phi_{i,b}$ and bottom thicknesses $x_{i,b}$ as well as top fatigue $\phi_{i,t}$ and top thickness $x_{i,t}$ within the same segment (correlation coefficients of approx. -0.95). This reflects the fact that thicker plates are less likely to result in a violation of the fatigue utilization. Interestingly, side fatigue $\phi_{i,s}$ and side thickness $x_{i,s}$ correlate much less (about -0.25). Similar observation can be made for stress (σ) and thicknesses, however, to a smaller degree

	$x_{3,b}$	$x_{3,s}$	$x_{3,t}$	$y_{3,b}$	$y_{3,s}$	$x_{4,b}$	$x_{4,s}$	$x_{4,t}$	$y_{4,b}$	$y_{4,s}$	z_b	z_s
$\sigma_{3,b}$	-0.62	-0.09	-0.14	0.15	0.00	-0.65	-0.07	-0.11	0.14	0.00	0.24	0.23
$\sigma_{3,s}$	-0.31	-0.18	-0.46	0.13	-0.07	-0.31	-0.15	-0.46	0.10	0.02	0.24	0.22
$\sigma_{3,t}$	-0.11	-0.12	-0.62	0.12	-0.08	-0.12	-0.16	-0.64	0.03	0.04	0.16	0.15
$\phi_{3,b}$	-0.95	-0.18	-0.20	0.07	0.00	-0.13	0.02	0.01	0.08	0.00	0.11	0.11
$\phi_{3,s}$	-0.65	-0.27	-0.55	0.09	-0.05	-0.08	0.02	-0.04	0.08	0.03	0.16	0.15
$\phi_{3,t}$	-0.14	-0.21	-0.96	0.08	-0.07	0.01	-0.07	-0.14	0.00	0.05	0.06	0.05
$\beta_{3,b}$	-0.66	-0.05	-0.01	-0.48	0.00	-0.34	-0.01	-0.03	0.09	-0.01	0.14	0.13
$\beta_{3,s}$	-0.30	-0.69	0.01	0.02	-0.36	-0.29	-0.11	0.05	0.11	0.02	0.11	0.06
$\sigma_{4,b}$	-0.18	0.09	-0.07	0.16	-0.01	-0.67	-0.09	-0.12	0.15	-0.01	0.22	0.23
$\sigma_{4,s}$	-0.14	0.02	-0.13	0.11	-0.07	-0.35	-0.16	-0.49	0.11	0.02	0.24	0.24
$\sigma_{4,t}$	-0.08	0.04	-0.13	0.07	-0.07	-0.14	-0.12	-0.67	0.02	0.03	0.17	0.18
$\phi_{4,b}$	-0.13	0.06	-0.04	0.11	0.00	-0.95	-0.20	-0.22	0.08	0.00	0.12	0.13
$\phi_{4,s}$	-0.08	0.03	-0.11	0.09	-0.04	-0.66	-0.25	-0.58	0.06	0.03	0.15	0.15
$\phi_{4,t}$	0.02	0.04	-0.16	0.04	-0.06	-0.16	-0.25	-0.96	-0.05	0.03	0.07	0.05
$\beta_{4,b}$	-0.13	0.11	-0.06	0.12	-0.01	-0.66	-0.04	-0.11	-0.45	-0.01	0.11	0.12
$\beta_{4,s}$	-0.06	-0.02	-0.05	0.04	-0.01	-0.34	-0.68	0.01	0.05	-0.36	0.02	-0.03

Fig. 3 Correlation matrix of the inputs (columns) and outputs (rows) for segments 3 and 4

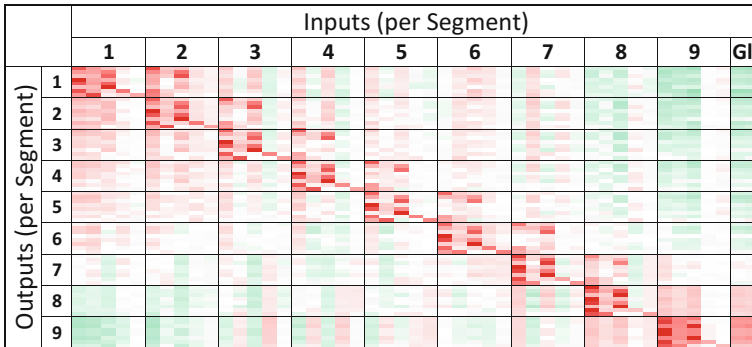


Fig. 4 Correlation matrix between the inputs (columns) and the outputs (rows) of the BTB. The order of the inputs and outputs per segment is the same as in Fig. 3

(correlations up to -0.67). While buckling (β) also correlates with the thicknesses, the more interesting observation is the correlation with the number of stiffeners (y). The latter is in accordance with the fact that the stiffeners are mostly used to avoid buckling. The stiffener types z only correlate to a small degree with the outputs.

Another aspect that can be observed in Fig. 3 is that the outputs of a segment generally correlate to the inputs of that same segment, but also to the inputs of the next segment to a smaller degree, e.g. the stress of segment 3 correlates with the thicknesses of segment 4 (up to -0.65). The inputs of the previous segment, however, have only very small correlation coefficients to the current segment (up to -0.18). Those aspects can be observed globally over all segments, as shown by the correlation matrix in Fig. 4. The reddish block diagonal shows the high correlations between inputs and outputs of the same segments. The lighter reddish blocks just

Table 1 Inputs and outputs for data analysis and surrogate modeling grouped by segments

Sample	Input					Output			
	Seg ₁	Seg ₂	...	Seg _n	Globals	Seg ₁	Seg ₂	...	Seg _n
#1	$\mathbf{x}_1, \mathbf{y}_1$	$\mathbf{x}_2, \mathbf{y}_2$...	$\mathbf{x}_n, \mathbf{y}_n$	\mathbf{z}	$\sigma_1, \phi_1, \beta_1$	$\sigma_2, \phi_2, \beta_2$...	$\sigma_n, \phi_n, \beta_n$
#2	$\mathbf{x}_1, \mathbf{y}_1$	$\mathbf{x}_2, \mathbf{y}_2$...	$\mathbf{x}_n, \mathbf{y}_n$	\mathbf{z}	$\sigma_1, \phi_1, \beta_1$	$\sigma_2, \phi_2, \beta_2$...	$\sigma_n, \phi_n, \beta_n$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

right to the main diagonal are the correlations between the next and current segments. These observations help in selecting (and thus minimizing) the relevant variables for training surrogate models, as discussed in more detail in the following section.

Lastly, we observed that the outputs are highly correlating with each other (data not shown in the paper), e.g. buckling bottom correlates with stress bottom. This offers the opportunity that surrogate models are trained not only based on the “regular inputs” (thicknesses and stiffeners), but also based on other statics constraints. For example modeling the stress with fatigue as additionally allowed input.

Variable Selection and Data Preprocessing

We first discuss the selection of an appropriate set of input variables to train the surrogate model. Generally, the more selected inputs, the higher the probability that all relevant information is available to generate a model that approximates the structural analysis well (i.e. has a high accuracy). However, including too many variables makes the training of the model more difficult and the chance is higher that certain variables are redundant or irrelevant (i.e. carry no or very little additional information). Thus, selecting a small set of relevant variables is an important task.

Grouping the thickness and stiffener variables per segment i , we define $\mathbf{x}_i = (x_{i,b}, x_{i,s}, x_{i,t})$ and $\mathbf{y}_i = (y_{i,b}, y_{i,s})$. Similarly, for the statics utilization constraints, we define for stress $\sigma_i = (\sigma_{i,b}, \sigma_{i,s}, \sigma_{i,t})$, fatigue $\phi_i = (\phi_{i,b}, \phi_{i,s}, \phi_{i,t})$ and buckling $\beta_i = (\beta_{i,b}, \beta_{i,s})$. These inputs and outputs are listed in Table 1, where each row correspond to one sample that was evaluated with the structural analysis tool.

Based on the results of Section “Data Analysis”, we define which input variables are selected for training the $8n$ models (one per output variable per segment). Besides the option of using all inputs to model an output of a specific segment, the analysis of the correlation matrix of the inputs and outputs reveals that especially the variables of the current segment i as well as the next segment $i + 1$ are good candidates to be used as inputs. Thus, we have the following options for selecting the input variables:

- variables from all segments plus the global variables: $5n + 2$ selected inputs, e.g. for a 9-segment boom 47 selected inputs

Table 2 A single sample (corresponding to one row in Table 1) is aligned into multiple data rows

Sample	Input				Output Seg_i
	Seg.No.	Seg_i	Seg_{i+1}	Globals	
#1	1	x_1, y_1	x_2, y_2	\mathbf{z}	$\sigma_1, \phi_1, \beta_1$
#1	2	x_2, y_2	x_3, y_3	\mathbf{z}	$\sigma_2, \phi_2, \beta_2$
⋮	⋮	⋮	⋮	⋮	⋮
#1	n	x_n, y_n	-	\mathbf{z}	$\sigma_n, \phi_n, \beta_n$
#2	1	x_1, y_1	x_2, y_2	\mathbf{z}	$\sigma_1, \phi_1, \beta_1$
#2	2	x_2, y_2	x_3, y_3	\mathbf{z}	$\sigma_2, \phi_2, \beta_2$
⋮	⋮	⋮	⋮	⋮	⋮

- variables of the current segment i plus the global variables: $5 + 2 = 7$ selected inputs
- variables of the current segment i and the next segment $i + 1$ plus the global variables: $2 \cdot 5 + 2 = 12$ selected inputs (Note that for the last segment there are only $5 + 2 = 7$ inputs, because there is no next segment)

Due to the structure of the BTB-problem, not only the inputs can be reduced, but also the number of models that need to be trained. In the elaborations above, $8n$ models were used (for each segment i , one model for each statics constraint). One option to reduce the number of models would be to create only one model that predicts the overall violation of the statics constraints, e.g. in form of the sum of all values larger than one. However, such a model would be difficult to train and to interpret because information is lost due to the aggregation. Instead, we opted for creating a single model per statics constraint that can predict the output for all segments, i.e. one model for fatigue bottom instead of one fatigue bottom model per segment. Thus, only 8 models need to be trained. To still be able to capture potential differences among the segments, the segment number is used as an additional input variable.

To facilitate training one model per output, the data is restructured to yield the structure shown in Table 2. In essence, the data is aligned per segment so that each new row contains outputs of a single segment. By splitting the data this way, a single evaluation of a boom with n segments yields n data rows of $5 + 2$ inputs and 8 outputs. Optionally, the inputs of adjacent segments can be included (potentially $5n + 2$ inputs). Including the input of the next segment (as shown in Table 2), we obtain $n - 1$ data rows of $2 \cdot 5 + 2$ input variables (for all segments except the last one) and one data row of $5 + 2$ input variables (for the last segment).

This segment-based data-splitting has two advantages: First, we only need to create a total of 8 models, one for each statics utilization, instead of $8n$ models. Second, we obtain more training data from a single evaluation. Thus, splitting the data by segments should speed up and simplify the modeling process.

However, when the inputs of the next segment are included, the last segment must be handled differently, because there is no next segment. We use a “dummy next segment”, where the thicknesses, number of stiffeners and stiffener types are set to zero.

Modeling Procedure

The selection of the modeling procedure is a central task, influenced by several factors. In order to be able to discuss surrogate models with domain experts and to potentially gain valuable insights in the connection between the input and output variables, we use white-box models in the form of mathematical expressions describing the relations between these variables. Obtaining such expressions also enables simplifying (e.g. by deleting expressions with no/low impact) and fine-tuning (e.g. optimizing the numeric constants computationally (Kommenda et al. 2013)) in a post-processing step. One simple white-box model is linear regression, which is, however, too restricted because we expect non-linear relations. Thus, Symbolic Regression (SR) was chosen as an extension of linear regression. We generate SR models using genetic programming (Koza 1992), and ALPS (Hornby 2006) and OSGA (Affenzeller and Wagner 2005) in combination with constant optimization (Kommenda et al. 2013). In online surrogate-assisted optimization (Forrester and Keane 2009; Wang and Shan 2007), (computationally cheap) black-box modeling techniques (e.g. random forests) can be used instead.

To group the models of each constraint, we introduce the notation $\sigma = (\sigma_b, \sigma_s, \sigma_t)$, where σ_b , σ_s and σ_t denote the utilization with regard to stress on the bottom, side and top, respectively. Similarly, $\phi = (\phi_b, \phi_s, \phi_t)$ and $\beta = (\beta_b, \beta_s)$ are defined.

Initially, we trained models for the 8 constraints σ , ϕ and β using the segment-based data-splitting approach with the segment number, the current and next segment as well as the global variables as input (cf. Section “[Variable Selection and Data Preprocessing](#)” and Table 2). We then extended the set of input variables to also allow other statics constraints as inputs, as suggested in the latter part of the data analysis in Section “[Data Analysis](#)”. Interestingly, only fatigue could be modeled well without other statics constraints as input variables; the accuracy of the models for stress and buckling was significantly lower without the extended set of input variables. Thus, we allowed the fatigue as extended input for modeling the stress. For modeling buckling we allowed fatigue and stress. Thus, we obtain the following models

$$\widehat{\phi} = f(i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_{i+1}, \mathbf{y}_{i+1}, \mathbf{z}) , \quad (13)$$

$$\widehat{\sigma} = f(i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_{i+1}, \mathbf{y}_{i+1}, \mathbf{z}, \phi) \text{ and} \quad (14)$$

$$\widehat{\beta} = f(i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_{i+1}, \mathbf{y}_{i+1}, \mathbf{z}, \phi, \sigma) , \quad (15)$$

where $\widehat{\phi}$, $\widehat{\sigma}$ and $\widehat{\beta}$ are the estimated values of σ , ϕ and β , respectively, i , \mathbf{x}_i , \mathbf{y}_i , \mathbf{x}_{i+1} , \mathbf{y}_{i+1} , \mathbf{z} the regular inputs and ϕ , σ the statics constraints as extended inputs.

After modeling, the statics constraints that are used as extended inputs – which are not present when estimating a novel boom configuration – must be estimated using the other models. This implies that circular dependencies among the models are not allowed. To distinguish between the models above (where the statics constraints are used from the training dataset) and the model where the statics constraints are estimated, we introduce the model variants

$$\widehat{\sigma}^* = f(i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_{i+1}, \mathbf{y}_{i+1}, \mathbf{z}, \widehat{\phi}) \text{ and} \tag{16}$$

$$\widehat{\beta}^* = f(i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_{i+1}, \mathbf{y}_{i+1}, \mathbf{z}, \widehat{\phi}, \widehat{\sigma}^*) \text{ ,} \tag{17}$$

where the estimated values of the statics constraints are used instead. In those models, the model errors accumulate, meaning that the overall model accuracy will be lower than in the models using the original values from the dataset.

As an additional step for models that use extended inputs, we incorporate the models of these extended inputs into the outer model. In other words, for a model \hat{a} that use other statics constraints b as extended input variables, we incorporated the models \hat{b} directly into the model \hat{a} by replacing each occurrence of these variable b by its model \hat{b} . For example, the stress model $\widehat{\sigma}^*$ actually includes the whole expression of the fatigue model $\widehat{\phi}$. This makes the handling of models simpler (the models are independent of each other), and enables fine-tuning (simplifying and constant optimization) of the whole model. After fine-tuning this larger model, the accuracy can be higher than when simply executing the models sequentially and passing the results to the dependent models. In the following results (Section “[Results](#)”), the models marked with * are the combined ones with fine-tuning applied.

Results

In this section, we present and discuss the modeling results for the 8 utilization constraints: bottom/side/top stress (σ), bottom/side/top fatigue (ϕ) and bottom/side buckling (β). First, we present an overview of all models and discuss the accuracy of the models and the impact of the input variables. Second, we discuss a single model (fatigue bottom) in detail, to show how the model can be used to gain insight into the general behavior of the BTB.

Models Overview

In this section, an overview of the results of all models is presented. Table 3 contains the models’ length in terms of number of nodes and the model’s accuracy in terms

Table 3 Accuracy of the model in terms of Pearson R² and Mean Absolute Error (MAE)

Model	Length	Pearson R ²		MAE (in 10 ⁻³)		Model	Length	Pearson R ²		MAE (in 10 ⁻³)	
		Training	Test	Training	Test			Training	Test	Training	Test
$\hat{\phi}_b$	20	0.999	0.999	1.812	1.830						
$\hat{\phi}_s$	32	0.994	0.994	4.052	4.052						
$\hat{\phi}_r$	22	0.999	0.999	1.760	1.762						
$\hat{\sigma}_b$	34	0.991	0.990	4.506	4.547	$\hat{\sigma}_b^*$	73	0.992	0.991	4.182	4.315
$\hat{\sigma}_s$	44	0.971	0.970	9.307	9.338	$\hat{\sigma}_s^*$	92	0.972	0.972	8.780	8.760
$\hat{\sigma}_r$	21	0.982	0.981	7.875	8.037	$\hat{\sigma}_r^*$	46	0.989	0.988	6.077	6.154
$\hat{\beta}_b$	35	0.974	0.978	17.42	17.87	$\hat{\beta}_b^*$	109	0.970	0.975	17.91	18.07
$\hat{\beta}_s$	34	0.964	0.965	25.70	26.34	$\hat{\beta}_s^*$	176	0.976	0.976	20.91	21.49

of the Pearson’s R² and mean absolute error. All models fit the data well, with the models for fatigue and stress generally having a very high accuracy, and models for buckling having a lower though still high accuracy. The high accuracies for all models indicate that, even though the underlying calculations of the statics analysis tool are complex, these constraints can indeed be approximated well with simple models.

Table 4 contains the *variable impacts* for all models. The impact of a variable describes how much the accuracy of a model (i.e. the Pearson R²) would decrease if this variable is not available (similar to node-impacts in Affenzeller et al. (2014)). To simulate the lack of a variable, the values of that variable are shuffled to break their relation to the target variable. A positive impact means that the quality of the model would decrease without the variable and vice versa for negative values. For example, a variable impact of 0.8 means that the model’s accuracy would be reduced by 0.8 (in terms of R²), indicating that this variable is quite important.

For all models, the segment number *i* has a very high impact, indicating that the constraints behave differently per segment. For final models with dependent models already incorporated (marked with *), the thicknesses always have a high impact, which is not surprising considering that thicker plates should result in more robust booms. The stiffener types are hardly used by any model, indicating that it is not important which type is used. The number of stiffeners of the next segment is not used by any model, thus it is not included in the table.

Interestingly, the models for fatigue only use the thicknesses of the current segment plus the segment number and still yield the highest accuracy. This indicates that the fatigue is influenced very directly and only by the thicknesses of the plates.

The models for the stress use the thicknesses of the current and the next segment, i.e. the stress constraints could be more likely to be violated if the subsequent segment is very heavy due to the subsequent plates’ thickness. Another interesting aspect is that the models for stress that are still using fatigue as input ($\hat{\sigma}$) do not use the thicknesses of the current segment, instead they depend heavily on the fatigue.

Table 4 Variable impacts of the inputs on the models. No entry means that the variable was not used in the model. Crossed out variables were not available as an input

Model	i	$x_{i,b}$	$x_{i,s}$	$x_{i,t}$	$y_{i,b}$	$y_{i,s}$	$x_{i+1,b}$	$x_{i+1,s}$	$x_{i+1,t}$	z_b	z_s	$\phi_{i,b}$	$\phi_{i,s}$	$\phi_{i,t}$	$\sigma_{i,b}$	$\sigma_{i,s}$	$\sigma_{i,t}$
$\widehat{\phi}_b$	0.67	0.81	0.08	0.04								/					/
$\widehat{\phi}_s$	0.74	0.56	0.12	0.38								/					/
$\widehat{\phi}_t$	0.77	0.03	0.08	0.75								/					/
$\widehat{\sigma}_b$	0.87						0.71	0.11	0.02			0.79					/
$\widehat{\sigma}_s$	0.63						0.69	0.19	0.48				0.55	0.10			/
$\widehat{\sigma}_t$	0.95							0.16	0.66					0.86			/
$\widehat{\sigma}_b^*$	0.89	0.51	0.06	0.03			0.68	0.10	0.02								/
$\widehat{\sigma}_s^*$	0.78	0.18	0.09	0.31			0.69	0.19	0.47								/
$\widehat{\sigma}_t^*$	0.87	0.02	0.06	0.44				0.12	0.74								/
$\widehat{\beta}_b$	0.29	0.79			0.50					0.02					0.35		
$\widehat{\beta}_s$	0.31		0.66			0.17						0.06	0.05		0.63		0.06
$\widehat{\beta}_b^*$	0.83	0.92	0.03	0.00	0.51		0.19	0.02		0.02							
$\widehat{\beta}_s^*$	0.97	0.37	0.74	0.02		0.18	0.59	0.12	0.01								

The buckling models are the only models that use the number of stiffeners, which supports our current understanding of the BTB that the stiffeners are only required to counter buckling. Interestingly, buckling bottom and side also depend on the stress bottom in both cases. Summarizing, buckling is mainly dependent on the thickness bottom/top and the number of stiffeners; and indirectly dependent on the thicknesses of the next segment via the stress.

Modeling Fatigue Bottom

We demonstrate how the models can be used to learn interesting aspects about the BTB, using the following simplified model of fatigue bottom:

$$\widehat{\phi}_b = \frac{\mathbf{c}(i) + \exp(-0.094288x_{i,t}) 1.9923 - 1.1311}{1.5519x_{i,b} + 0.79312x_{i,s} + 2.0162} + 0.017964 \quad (18)$$

$$\mathbf{c} = (7.7098, 7.7719, 7.8131, 7.8282, 7.7843, 7.6066, 7.1214, 5.8482, 2.6857)$$

The main impacts in the model for fatigue bottom in Eq. (18) are the terms covering the segment-dependent vector $\mathbf{c}(i)$ and the bottom thickness $1.5519x_{i,b}$. These high impacts are also in line with the impact factors in Table 4.

Instead of using the segment number i directly as an integer, a vector \mathbf{c} is used where each entry corresponds to one segment. The values for segments 1 to 7 are significantly higher, which indicates that those segments are more likely to violate the constraint (assuming the other variables are fixed). The thickness bottom and side are both in the denominator, meaning that they are inverse proportional to the fatigue. The thickness top is included in the model within the term $\exp(-x)$, also meaning that it is decreasing with higher values. Both facts combined imply that higher thicknesses reduce the likelihood that the fatigue constraint is violated. The models for fatigue side and top (not discussed in detail here) behave similarly.

The models for the other constraints are not discussed here in detail. All models can be found online at <http://dev.heuristiclab.com/AdditionalMaterial>.

Summary and Discussion of the Results

The results show that the structural analysis of the BTB has large potential for applying surrogate modeling. This suggests that, after further investigation and discussion with domain experts, the expensive simulations – which are currently used to determine the utilization of stress, fatigue and buckling – could be replaced by much faster models. Additionally, valuable information can be obtained by analyzing the models in more detail, as demonstrated with the simplified model for fatigue bottom.

As a next step, to solve the BTB optimization problem, the created models could be used in optimization approaches that potentially require a large number of solution evaluations because the mathematical expressions of the surrogate models can be evaluated within microseconds. When accounting for the overall runtime to solve the BTB problem, however, the efforts spent to sample the data and learn the surrogate model must also be accounted for. Currently, the most calculation intensive part is creating the samples for training the model, which can take several days up to a few weeks, depending on the computational resources. Modeling (including data preprocessing and model postprocessing) can be done within a few days.

One drawback of the presented models is that they might not be generally applicable for all customer requirements, because the sampled data were based on one boom with a fixed length and a fixed set of load-cases. For other settings, sampling and modeling would have to be redone. However, the general behavior of the models (e.g. in terms of thicker plates being inverse proportional to violations in the utilizations) are expected to be similar. In that case, potentially less samples are required and the existing models are just adapted, i.e. their numeric constants recalculated to fit the new data.

Conclusion

The paper presents the industrial use case of a box-type boom crane as potential benchmark for comparing optimization approaches. The BTB optimization problem is easy to understand and relatively simple to specify, with the exception of the structural analysis. For the goal of providing this use case to other researchers, future attempts include providing the implementation of the problem using the learned surrogate models as well as developing a simplified open-source version of the structural analysis tool.

As the second contribution of the paper, surrogate modeling for the runtime expensive structural analysis is presented, with the intention of replacing this expensive evaluation in optimization algorithms with the simpler surrogate model. Due to the structure of the BTB problem, in particular due to the segment-based partitioning, several strategies for surrogate modeling are possible, ranging from only selecting certain input variables (e.g. only variables of the current segment) to splitting the data based on the segments for reducing the number of outputs. Generally, the models capture the statics constraints very well in terms of their accuracy, suggesting that the expensive structural analysis tool can indeed be substituted by much simpler and faster models. Future directions with regard to surrogate modeling are to train more complex surrogate models that consider different load-cases and other characteristics of a boom as input data in order to be able to generalize the model for several booms.

The main avenue of continuing this work is to apply and compare different optimization methods, such as heuristic approaches, single solution-based algorithms as well as population-based optimization approaches, on the BTB problem. The goal is to investigate which optimization approaches are most promising for this kind of industrial use case.

Acknowledgements This work was carried out within the COMET K-Project #843551 “Advanced Engineering Design Automation (AEDA)” and the COMET K-Project #843532 “Heuristic Optimization in Production and Logistics (HOPL)”, both funded by the Austrian Research Promotion Agency FFG.

References

- Affenzeller M, Wagner S (2005) Offspring selection: a new self-adaptive selection scheme for genetic algorithms. In: Ribeiro DB, Albrecht DRF, Dobnikar DA, Pearson DDW, Steele DNC (eds) Adaptive and natural computing algorithms. Springer, Vienna
- Affenzeller M, Winkler SM, Kronberger G, Kommenda M, Burlacu B, Wagner S (2014) Gaining deeper insights in symbolic regression. In: Genetic programming theory and practice XI. Springer, Berlin, pp 175–190
- Affenzeller M, Beham A, Vonolfen S, Pitzer E, Winkler SM, Hutterer S, Kommenda M, Kofler M, Kronberger G, Wagner S (2015) Simulation-based optimization with HeuristicLab: practical guidelines and real-world applications. In: Applied simulation and optimization. Springer International Publishing, New York, pp 3–38
- Bishop C (2006) Pattern recognition and machine learning. Springer, Berlin

- Deb K (2010) Optimization for engineering design: algorithms and examples, 11th edn. PHI Learning Pvt. Ltd, New Delhi
- Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1):50–79
- Frank G, Entner D, Prante T, Khachatouri V, Schwarz M (2014) Towards a generic framework of engineering design automation for creating complex CAD models. *Int J Adv Syst Meas* 7(1 and 2):179–192
- HeuristicLab. <http://dev.heuristiclab.com/>. Accessed 03 Dec 2017
- Hornby GS (2006) Alps: the age-layered population structure for reducing the problem of premature convergence. In: Proceedings of the 8th annual conference on genetic and evolutionary computation. ACM, New York, pp 815–822
- Hubka V, Eder EW (1987) A scientific approach to engineering design. *Des Stud* 8(3):123
- Kommenda M, Kronberger G, Winkler S, Affenzeller M, Wagner S (2013) Effects of constant optimization by nonlinear least squares minimization in symbolic regression. In: Proceedings of the 15th annual conference companion on Genetic and evolutionary computation. ACM, New York, pp 1121–1128
- Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT press, Cambridge, MA
- Lehmn A, O'Rourke N, Hatcher L, Stepanski E (2005) JMP for basic univariate and multivariate statistics: a step-by-step guide, 1st edn. SAS Institute
- Liebherr-Werk Nenzing GmbH. <http://www.liebherr.com/en-GB/35267.wfw>. Accessed 06 Mar 2017
- Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. *J Eng Ind* 98(3):1021
- Roy R, Hinduja S, Teti R (2008) Recent advances in engineering design optimisation: challenges and future trends. *CIRP Ann-Manuf Technol* 57(2):697–715
- Sandgren E (1988) Nonlinear integer and discrete programming in mechanical design. In: Proceedings of the ASME design technology conference
- Wagner S, Kronberger G, Beham A, Kommenda M, Scheibenpflug A, Pitzer E, Vonolfen S, Kofler M, Winkler S, Dorfer V, Affenzeller M (2014) Architecture and design of the HeuristicLab optimization environment. In: Advanced methods and applications in computational intelligence, No. 6 in topics in intelligent engineering and informatics. Springer International Publishing, New York, pp 197–261
- Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. *J Mech Des* 129(4):370–380
- Zhang LL (2014) Product configuration: a review of the state-of-the-art and future research. *Int J Prod Res* 52(21):6381

Knowledge Objects Enable Mass-Individualization



Joel Johansson and Fredrik Elgh

Abstract Mass customization and product individualization are driving factors behind design automation, which in turn are enabled through the formalization and automation of engineering work. The goal is to offer customers optimized solutions to their needs timely and as profitable as possible. The path to achieve such a remarkable goal can be very winding and tricky for many companies, or even non-existing at the moment being. To succeed requires three essential parts: formally represented product knowledge, facilities to automatically apply the product knowledge, and optimization algorithms. This paper shows how these three parts can be supported in engineer-to-order businesses through the concept of knowledge objects. Knowledge Objects are human readable descriptions of formalized knowledge bundled with corresponding computer routines for the automation of that knowledge. One case example is given at the end of the paper to demonstrate the use of knowledge objects.

Introduction

Mass-customization (Hvam et al. 2008) is one of the most important competitive strategies in the current economy (Blecker and Abdelkafi 2006) and has been steadily growing for about three decades (Fogliatto et al. 2012; Silveira et al. 2001; Pine and Davis 1999). The reason for this rapid growth and its adoption by manufacturing industries is the great potential to improve customer value. The idea is to strive for a broad offer of products and at the same time ensure production efficiency.

J. Johansson (✉) · F. Elgh
Jönköping University, Box 1026, 55 111 Jönköping, Sweden
e-mail: joel.johansson@ju.se

F. Elgh
e-mail: fredrik.elgh@ju.se

Individualized products (also referred to as custom-engineered or one-of-a-kind products) include products that do not let them self be modularized into pre-defined modules. This is often due to the early customer decoupling point (Rudberg and Wikner 2004) which requires an engineer-to-order (ETO) approach instead of the common configure-to-order approach. This means that engineering has to be done in development, quotation preparation and order processing for every customer enquiry and order. The ETO process which allows products to be individually adapted to large variations in customers' specifications can be more or less formalized but includes similar engineering tasks for every customer enquiry and order (tasks may be added or cancelled from time to time).

The intention with the research presented in this paper is to enable ETO businesses to move towards fully automated ETO processes. In long terms it will enable ETO businesses to apply optimization algorithms to parts of or to the entire automated process making mass-individualization a reality; with the goal to offer customers individually optimized solutions to their specific needs timely and as profitable as possible.

To succeed with mass-individualization requires three essential parts namely formally represented product knowledge (further on referred to as formalized product knowledge), facilities to automatically apply the formalized product knowledge, and optimization algorithms.

To enable optimization algorithms to be applied to parts of, or to the entire, ETO process requires not only the development of optimization routines but also the automation of ETO sub-processes which are executed to evaluate the cost functions. To automate the ETO sub-processes requires the capturing and formalization of the engineering work in the ETO sub-processes. The optimization of ETO products are in other words resting on design automation, that in turn rests on formalized engineering knowledge.

Automated ETO sub-processes need to be executable by computers to be automated but they also need to be readable and comprehensible by humans to be maintained as market, technology and legacy change. A class of objects called Knowledge Objects was proposed by the authors of this paper (Elgh and Cederfeldt 2007; Johansson 2011; Elgh and Johansson 2014). In this work, Knowledge Objects are human readable descriptions of formalized knowledge bundled with corresponding computer routines for the automation of that knowledge (see Definition 1 in Section "Knowledge Objects"). This paper propose Knowledge Objects as fundamental building blocks for ETO processes to be formalized, automated and targeted for optimization algorithms.

These three parts: formalized product knowledge, design automation, and optimization are indeed their own research fields and much can be said about them. These fields are summarized in Section "Frame of Reference". Then the concept of knowledge objects is introduced in Section "Knowledge Objects". To demonstrate the use of knowledge objects a case example is described in Section "Case Example: Automated Weldability Analysis".

Frame of Reference

To enable product individualization where broad aspects of the product are taken into consideration requires a sound foundation of formalized product knowledge, design automation and optimization. Much can be said about these three topics. However, this section very shortly briefs the reader regarding them.

Formalized Product Knowledge

Engineering design problems are solved through iterations between synthesis and analysis phases. Design proposals developed through creative processes are evaluated based on requirements. During new product development processes much of the trial and error loops occur as a part of the learning of how to solve the problems connected with the product. The knowledge developed during these trials is referred to as product knowledge (Kennedy et al. 2008). When the product matures a base of tested solutions will emerge, a base that is reviewed on customer demanding the product for a different set of requirements. When such a base of solutions exists the synthesis phase gradually turns into a search for existing solutions that can be combined to solve new problems. Also, when the product matures the way of testing the product to requirements is formalized and can sometimes be skipped based on an inductive way of reasoning, i.e. based on experience it can be concluded that the new solution will fit. In the most mature state of an engineer-to-order product the processes for developing a new variant are well defined and based on user requirements that is configured so that the product knowledge can be utilized in synthesis and analysis phases to great extent.

There are methods and tools to capture and structure the product knowledge. One method for knowledge modelling, applicable in the domain of design automation systems, is the Systems Modelling Language (SysML) (Friedenthal et al. 2012). CommonKADS is a method to document and manage engineering knowledge (Schreiber and Akkermans 2000). It acts as a baseline for system development and research projects.

Product Variant Master is an operational tool to model and visualize a product family (Hvam et al. 2008). In general, a product family can be modelled in Product Variant Master as a Part-of structure which shows the components included in the product, and a Kind-of structure that shows the variants available.

Design Automation

Engineering problems are typically ill-structured (Simon 1973) because they lack well-defined ways of testing any suggested solution, the states of the problem are hard

to represent and the knowledge of the problems is hard to capture and represent. This fundamental properties of engineering problems has compelled the use of artificial intelligence in engineering design.

Knowledge based engineering (KBE) is a method to synthesize design proposals and has been defined as a technology based on the use of dedicated software able to capture and systematically reuse product and process engineering knowledge, with the final goal of reducing time and costs of product development by means of the following: Automation of repetitive and non-creative design tasks, and support of multidisciplinary design optimization in all the phases of the design process (Rocca and Tooren 2012). Traditionally KBE is based on knowledge-based-system from artificial intelligence where production rules are the fundamental carrier of the captured knowledge.

Case Based Reasoning (CBR) is another method to synthesize design proposals based on digitally stored experiences (referred to as cases) and reusing them in new situations (new cases). The method is based on four main operations: retrieve, evaluate (also referred to as reuse), revise, and retain (Agnar and Plaza 1994; Zhu et al. 2015). One advantage with CBR is that knowledge acquisition in CBR consist of a simple process of collecting examples and when comparing to KBE a way to short-cut a work-labor intensive process of developing formal knowledge.

In literature many examples can be found of how design automation has been applied to many parts of the product development process ranging from design synthesis through design evaluation and to production planning.

Optimization

In mathematics, optimization is the discipline concerned with finding inputs of a function that minimize or maximize the function value. The selection of values for design variables may be subjected to constraints (Pardalos and Resende 2002). Optimization in product development means finding the best solution among many feasible solutions (i.e. optimal solutions for the customers needs that can be produced with profit). Feasible solutions are those that satisfy all the constraints in the optimization problem. The best solution could minimize the cost of a process or maximizing the efficiency of a system (Arora 2015).

There exist a multitude of optimization algorithms but all optimization problems can mathematically be defined in a standardized way as presented by Arora (2015): Find an n -vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$ of design variables to *minimize* a cost function

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n) \quad (1)$$

subject to the p equality constraints

$$h_j(\mathbf{x}) = h_j(x_1, x_2, \dots, x_n) = 0; \quad j = 1 \text{ to } p \quad (2)$$

and the m inequality constraints

$$g_i(\mathbf{x}) = g_i(x_1, x_2, \dots, x_n) \leq 0; \quad i = 1 \text{ to } m \quad (3)$$

Through design automation, design proposals are generated and analysed and evaluated based on performance and production requirements. Finite-Element-Method is a widely used method for the evaluation which compared to data-base queries in CBR and rule firing in KBE is very computational expensive, i.e. generating design proposals takes fractions of the time necessary to evaluate them. This has led to the development of surrogate modelling or meta-modelling. Surrogate modelling (Jin et al. 2001; Forrester and Keane 2009) includes development of response surfaces based on design of experiments. The response surfaces are subsequently used as cost functions in optimization algorithms. The design variables in product development are typically continuous and discrete mixed together, which is hard to many optimisation algorithms. Another difficulty with optimization on ETO-processes is that f as product structures change, also the cost function changes. However, optimization has proven successful on many sub-processes of the ETO-process.

Knowledge Objects

In product development knowledge dictates what to do and what not to do in different situations. The knowledge that engineers use in product development processes appears in different kinds. In this context knowledge includes facts, rules and conditions. To enable mass-individualization it is necessary to automate the engineering activities included in the process of going from customer enquiry and order to design proposal further on to design analysis and evaluation and finally production release. That process, the ETO process, will include different activities from time to time and will change over long terms as new technologies emerge and market needs and legacy change. This requires the ETO process not only to be automated so that computers can execute the activities in a very flexible way but it also has to be described and represented so that humans can read and understand the process in order maintain it over long time. To achieve that it is here proposed to use knowledge objects as the fundamental base.

Definition 1 Knowledge Objects are bundles of human comprehensible knowledge representations and computer routines for the automated application of the represented knowledge.

A knowledge object with surroundings is visualized in Fig. 1. The knowledge object consists of an automation compartment and a description compartment.¹ The intention with the content within the automation compartment is mainly to be machine

¹Note that the knowledge representations not necessarily need to be physically stored within the knowledge object. The description compartment may consist of references to the representations as well. The user will get the feeling that description and automation compartments form a whole.

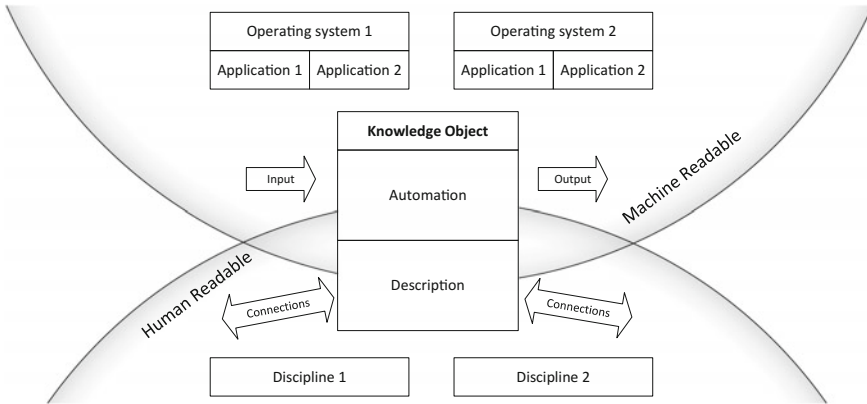


Fig. 1 A schematic view of a knowledge object and its surroundings

readable, even if it to some extent is interpreted by humans, this is illustrated by the arc labelled Machine Readable. Similarly, the intent of the description part is to be read and understood by humans even if managed by computers, this is illustrated by the arc labelled Human Readable. There exist some representation methods that are human readable and machine readable at the same time. Spread sheets can be examples of that. In the context of product development, the description compartment of a knowledge object consists of human comprehensible descriptions of:

- the piece of product knowledge represented by the knowledge object for one or several disciplines;
- where and when the piece of product knowledge is applicable;
- the computer routines in the automation compartment.

Descriptions of different knowledge objects may be interconnected enabling navigation of formalized knowledge for the product, this is illustrated by the two arrows labelled Connections in Fig. 1. Descriptions may also have internal connections so that it is possible to navigate the content (this is not illustrated in Fig. 1) The automation compartment of a knowledge object includes computer code for the automated application of the knowledge represented by the knowledge object for one or several operating systems and targeted software applications. Figure 1 indicates that the automation routines take inputs and produce outputs. These computer routines are used to automate the engineering activities in the ETO-process while the descriptions enable its maintenance. This section contains four subsections describing how formalized product knowledge can be represented by knowledge objects' description compartments, how the knowledge can be automatically applied through knowledge objects, how systems of knowledge objects are constituted and how optimization can be supported by knowledge objects and systems of knowledge objects.

Formalized Product Knowledge

A CAD-model is not itself knowledge. Nor is a snippet of programming code or an equation knowledge themselves. They are the result of human creativity and are representations of the result of applying human knowledge. In the context of product development these results define the product and we refer to them as design definitions further on (see Definition 4). CAD-models define what something is looking like and computer programming code and equations define how something is to be done, and are the result of applying engineering creativity through the product development process. However, more must be added to make a complete representation of product knowledge. That something includes descriptions about how to make use of the resulting design definitions, why these results are the way they are, what assumptions and simplifications were done when producing them, who produced them, when were they produced, why were they produced, and when are they valid. Available formats for making these descriptions includes text, mathematical expressions, diagrams, pictures, videos, audio, and three dimensional models.

The concepts of design descriptions were introduced by Elgh (2011) as a way to document product families of mass-individualized products. In that paper it is stated that:

The main focus of the Design Definition is the construction and the function of a process output object whereas the main focus of the Design Rationale is the argumentation and supporting descriptions unfolding and justifying the object design. Both the Design Definition and the Design Rationale provides essential meta-knowledge about the process output object and together they constitute the foundation for the Design Description.

This leads us to the following three definitions:

Definition 2 Design Definitions define what the product is and how it is manufactured.

Definition 3 Design Rationales include information regarding the purpose of the design, the reasons behind the design decisions and rejected design alternatives.

Definition 4 Design Descriptions are compositions of design definitions and design rationales that are representing the same piece of product knowledge.

The content of Design Definitions includes, by example, explanations of the overall product, its building blocks at different levels (e.g. product, assemblies, components, features and geometrical entities), relations between building blocks (e.g. functional structure and assembly sequence), parameters (input, internal and output), rules and tables describing the design space (Elgh 2011).

The content of design rationale includes information and links concerning aspects such as calculations, analyses, field test, underlying principles for design, assumptions, constraints, context, valid ranges of parameters and aspects for validity of rules, together with statements regarding what to consider when changing, ideas not yet

implemented and workarounds. A general object model to represent design rationales was developed by Elgh (2011) and further refined by Poorkiany et al. (2016) and is adapted as a means to represent design rationale within knowledge objects.

The description should include a categorization of the represented knowledge defining what discipline it belongs to. Ownership should also be assigned to user roles so that right person may be contacted.

It is important to describe the computer routines that automates the application of the formalized knowledge. Such description should include what platform the routines target. If the routines automate software through APIs it should contain information regarding valid versions of the API. Input and output parameters on the low level function calls should be described with explanations and valid ranges of the values. Any naming convention of the parameters should be attached. If named algorithms are used, these names should be stated with references to literature. Simplifications and known bugs should be listed. It can sometimes be useful to add a precision value to allow knowledge-bases to contain overloaded knowledge objects (Johansson 2007).

Design Automation

Knowledge change over time and a knowledge-base needs to be flexible so that pieces of knowledge can be easily added, updated, or deleted without disrupting the operation of the system. To make a system flexible in that sense, all parts of the system should be autonomous. Hence, to make the knowledge base flexible, all the chunks of knowledge must be as autonomous as possible. A procedural programming approach is not practical in this case, since changing the knowledge base would mean changing, compiling, and subsequently distributing the programming code. Instead, a declarative way of implementing the knowledge-base has proven to be fruitful.

In a declarative system, the knowledge-base is separated from the functions that make use of the knowledge-base. Consequently, changing the knowledge-base in a declarative system can be performed in a plug-and-play manner. Since the knowledge that engineers use when designing products and developing production tooling is highly connected to various tasks and concepts, the use of object-oriented knowledge bases has proven to be successful. Object-oriented programming offers the possibility to develop highly flexible software. The main idea with objects in object oriented programming is to store related data and functions together. A class of objects called knowledge objects was proposed by the authors of this paper (Elgh and Cederfeldt 2007; Johansson 2011; Elgh and Johansson 2014). The focus in previous works with knowledge objects has been on the automation part which contains a list of input parameters, a list of output parameters, and a method for processing input parameters to output parameters. It has been said that other fields may be added to a knowledge object, for example constraints, owner, categories, precision, and comments. However, these additional fields now belong to the description compartment rather than the automation compartment of the knowledge object.

When implementing knowledge objects, they should be defined in a way that makes them autonomous and the methods used to process information should preferably be in external software applications. The benefits of developing knowledge objects that are autonomous using common wide-spread applications as methods are two-fold: the knowledge can be used manually without the design automation system, and it is easy to find people skilled enough to use the very same knowledge the design automation system does - it makes the knowledge more human-readable and the synchronization of the description part easier. However, it is often necessary to standardize the way the process information is added in the selected applications and make the information richer than is normally seen. A spread sheet with only data and formulas in the cells does not make a complete knowledge object, only the automation part. Filling it with comments and instructions explaining the content would complete the knowledge object.

Systems of Knowledge Objects

Figure 2 shows a diagram with all components needed to form a complete system based on knowledge objects. A **Knowledge Domain** is a set of **Knowledge Objects** and **GlobalParameters**. Each knowledge object has two sets **LocalParameter** one for inputs and one for outputs. **GlobalParameter** and **LocalParameter** both inherit the class **Parameter** and are mapped in a one-to-many relation so that one **GlobalParameter** can have several representations on a local level as **LocalParameter**. This arrangement makes the implemented system flexible to changes, enabling the plug and play of knowledge objects. Each knowledge object is also associated with an **ExecutionMethod** which automates a piece of software to transforms the specified inputs into outputs.

The execution of the knowledge objects within the knowledge domain is scheduled by an **InferenceEngine**. The inference engine arranges the automated knowledge in the knowledge objects in an executable order based on the input and output parameters of the automation part of the knowledge objects². This can be done prior to execution or dynamically during the execution of the system. Two main types of search-based inference engines exist: forward and backward-chaining. A forward-chaining (also called data-driven) mechanism uses the information initially presented to execute all applicable knowledge objects. The method has two steps. The inference engine searches for knowledge objects with all input parameters known, and applicable in the current state. It then selects one of the found knowledge objects to execute the execution method defined in that knowledge object to retrieve output parameters using the input parameters. When the method has run, the stock of known parameters is updated, and a new search for executable knowledge objects is initiated. The process proceeds until no more knowledge objects can be executed. A

²A human reader of the knowledge base may also follow these paths of automation because they make sense.

backward-chaining (also called goal-driven) inference mechanism instead is initialized by asking for desired output values. It then searches for what knowledge objects have to be executed to find the desired output values. The knowledge objects are the fired to retrieve the outputs. Backward-chaining has not been implemented for Knowledge Objects.

It is also possible to navigate the knowledge base through the links between the describing part of the knowledge objects. An **Explainer** class can be added to the knowledge domain that manages description part of knowledge objects to enhance the navigation of the represented product knowledge during and after execution of the system.

Knowledge objects can be arranged in systems with different levels. This is achieved by encapsulating **KnowledgeDomains** as **KnowledgeObjects**. Having systems of knowledge objects in this way makes it possible to model the knowledge to match the structure of the product that is targeted for mass-individualization so that sub-processes of the ETO process target sub-levels of the product structure.

Optimization

An optimization algorithm exposes design variables, cost functions and constraints (see Section “[Optimization](#)”). In order to facilitate functions for optimization based on knowledge objects a **Loop** class was added to the **Knowledge Domain**, see Fig. 2. The **Loop** class is a superclass that can be specialized as any optimization algorithm. This is indicated in Fig. 2 by the two inheriting classes **Exhaustive Search** and **Opti-**

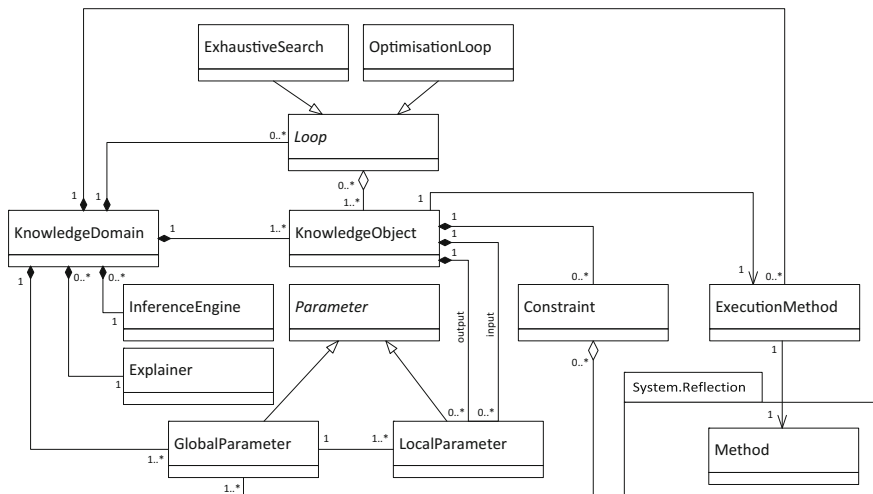


Fig. 2 Design automation system based on knowledge objects

mizationLoop. A **Loop** object contains a list of **Knowledge Objects** which implicitly yields a list of involved input, intermediate and output **Parameters**, which are the design variables (see Section “**Optimization**”). The list of knowledge object also implicitly defines a set of constrains that have to be considered by the loop. A **Loop** object also contains fields to support the optimization algorithm such as scheduling test designs, iteration counter, facilities to store data for each iteration and to check the status of the contained knowledge objects and the loop itself, whether it is executable or not. Two specialized loop classes have been developed and tested on knowledge objects, exhaustive search which is based on factorial studies and the Complex algorithm (Box 1965; Guin 1968) which is a non-gradient based optimization algorithm where search direction is constructed on n-1 worst designs in a population and has been applied to a variety of engineering problems. The Complex algorithm is suitable for design problems as it is robust when the design parameters are a mixture of discrete and continuous domains.

The cost function of the optimization algorithm is constituted by the automation part of the knowledge objects included in the **Loop** and maximisation or minimisation targets can be applied to any of the output parameters from the loop. It is possible to put equality or in-equality constrains on any of the design parameters in the loop.

Since simulations may be computationally expensive and time-consuming, each optimization run could entail hours of computation time to find a set of optimal solution. It is thus important that the number of cost function evaluations required to find the optimal solution is as low as possible. Especially multi-objective optimization of engineering problems using complex method or evolutionary algorithms require many evaluations of each objective within the design space, leading to a large number of simulations runs. Meta-model based multi objective optimization is most commonly used, where the optimization algorithm is applied on one or several meta-models representing the actual simulations. The basic idea of meta-modelling is to create a simplified approximation function of the real model (simulations) in some sampling points within the design space. Meta-models can be developed for each **KnowledgeObject** or globally for a **Loop** object.

Case Example: Automated Weldability Analysis

Real cases of mass-individualized products based on optimization are yet to be seen. However, many companies move towards automation of engineering processes and in this section we will look at one case example that demonstrates the description and the looping facilities of knowledge objects. The example system implements knowledge objects based on the software Howtimation Suite (Johansson 2015) which is based on the Microsoft .net platform and implements knowledge objects as described in this article. The knowledge objects for weldability analysis were developed by Pabolu et al. (2016) and was applied to jet-engine components, see Fig. 3 for an example.

Rules for evaluating weldability were extracted from various sources such as scientific literature, welding handbooks, weld equipment operation manuals, industrial

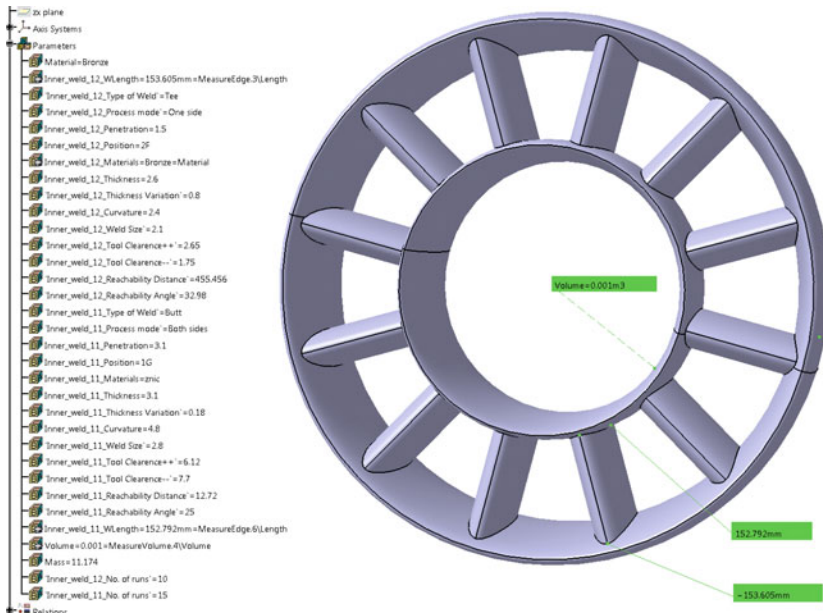


Fig. 3 Illustrative CAD-model of part from the case company. To the left the parameter tree from the CAD-system is visible with named parameters (Pabolu et al. 2016)

standards or best practices. Usually the available rules were documented in the form of PDF-documents. Each of the captured rules was then described and automated in spread sheets in Microsoft Excel³. Spread sheet is a suitable format to represent product knowledge as knowledge objects as the human readable description of the knowledge is put close to the automation routines (the formulas in the work-sheet). The layout of the spread sheets was standardized so that each has a description section and an automation section. The description section contains information regarding the purpose of the formulas in the automation section and also hyperlinks to underlying PDF-documents for further references. Each of the spread sheets was then wrapped as **KnowledgeObjects**. There are in total 17 knowledge objects and 58 parameters in the knowledge domain for weldability analysis, see Table 1 for a complete list of automated tasks and Fig. 4 for a graphical overview of the knowledge domain as it looks in the Howtotation Suite.

The weldability of each design from the design of experiments is evaluated through the knowledge objects which are executed in an order dynamically determined by the inference engine. As an example: if the material is not suitable to weld or if the thickness is out of feasible range, then the result becomes “not OK” for the thickness feasibility check. The reason will be put in an output spread sheet, which is

³In the case example all Knowledge Objects are targeting Spread Sheets. However, Knowledge Objects can target CAD-models, databases, constraint solvers, FEM-simulations, or practically any software. The case example was selected to demonstrate the looping functionality.

Table 1 Knowledge objects in the weldability analysis system

Task	Inputs	Outputs
Retrieve parameter values from CAD-model	0	14
Check feasibility for combination of materials	2	3
Check feasibility for weld position	2	3
Check the weld method compatibility with weld type	2	3
Check weather weld method can do single sided and/or double sided welds	2	3
Check whether the weld gun is fit to make the weld in vertical direction	2	3
Check whether the weld gun is fit to make the weld in angular direction	2	3
Estimate the reachability of the weld gun to the weld location	2	3
Estimate the reachability of the weld gun to the weld location	2	3
Check whether the plate thickens is feasible for the weld method	4	3
Check whether the weld will be thick enough	4	3
Check the compatibility of plate thicknesses and weld method	4	3
Check the weld size compatibility with weld method	4	3
Cancel welding method if turns are to tight	4	3
Calculate welding cost	4	1
Summarize all results in spread sheet and put out weldability index	43	1

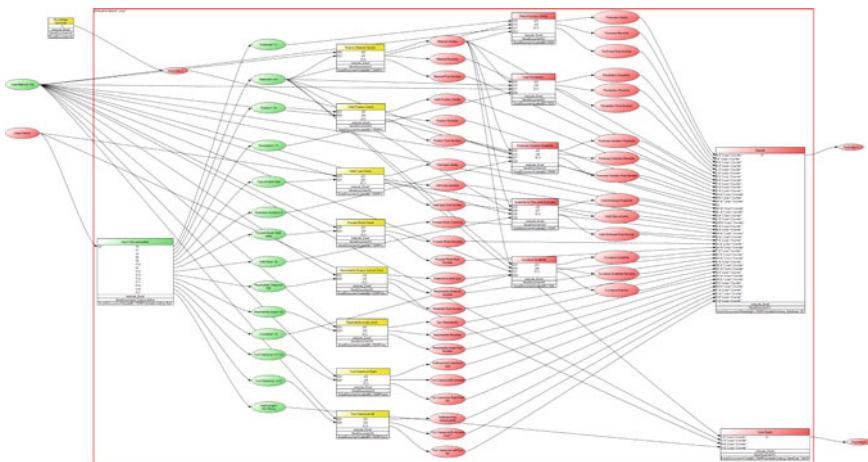


Fig. 4 Screen shot from the Howtomatic Suite of the knowledge domain for weldability analysis

also wrapped as a Knowledge Object. The reasons for failure can be: ‘Material is infeasible for weld’ or ‘The plate thickness is out of the feasible range’ which is also indicated in the output spread sheet. If the thickness is feasible then a difficulty-value is calculated based on the captured knowledge. After the analysis of the evaluation rules, the results, reasons and the difficulty levels, which are based on given inputs are transferred to the output spread sheet. The output spread sheet consists of

weldability index, cost calculation and a sustainability value for each design from design of experiments. These are further summarised in to a single summary sheet in a defined format which consists of weldability index, weld cost, weld feasibility, energy consumption and the environmental impact based on each weld joint. In case of manufacturing conflicts between the design and welding, a negative welding index value indicates infeasibility.

The weldability analysis knowledge domain contains one **Loop** object of the type **ExhaustiveSearch**. The loop contains 16 of the 17 knowledge objects having 3 input, 53 intermediate and 2 output parameters. The input parameters includes Weld Method, Weld Position, and Weld Length. Output parameters includes Weldability Index and Welding Cost. The knowledge object not included in the loop is a knowledge object with no input parameters which makes it an initialization object, i.e. it will execute first (listed first in Table 1). That knowledge object connects to a running instance of CATIA and retrieves current welding positions and welding lengths of the active CATIA-model. That information is retrieved from parameters calculated within the CAD-system and is subsequently submitted to other knowledge object in the knowledge domain.

When executing the system (from CATIA) the information in specially named parameters in the CAD-model will be retrieved. Subsequently the exhaustive search will start to loop through all possible variations of welding methods for each weld in the component. For each iteration a new row with all important results, such as weldability index and welding cost for the component, will be written in an output Excel spread sheet. The combination with the best welding index will be selected. However, this may be the most costly welding method and engineers can take other decisions based on data filtering functions in Excel. In future multi-objective optimization will be applied to also automate this last step.

Conclusions

This paper shows that Knowledge Objects enable mass-individualization as they provide a foundation for three important keys for engineer-to-order companies. First, they enable capturing of corporate knowledge regarding products and their related processes as Descriptions. Second, they carry computer routines for the automation of the captured knowledge bundled together with the descriptions. Finally, through the general looping functionality they support optimization of synthesis and analysis phases in product development including manufacturability assessments. These three parts together makes it possible for engineer-to-order companies to offer customers optimized solutions timely and with high profit. That is mass-individualization.

Future work includes development of working methods and tools so that engineering knowledge can be captured, represented and managed as knowledge objects throughout the entire product development process. Future work also includes the application of meta-modelling and more sophisticated optimization algorithms to knowledge objects.

References

- Agnar A, Plaza E (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Commun* 7(1):39–59. <https://doi.org/10.3233/AIC-1994-7104>
- Arora RK (2015) Optimization: algorithms and applications. CRC Press, Boca Raton
- Blecker T, Abdelkafi N (2006) Mass customization: state-of-the-art and challenges. Springer, Boston, pp 1–25. https://doi.org/10.1007/0-387-32224-8_1
- Box MJ (1965) A new method of constrained optimization and a comparison with other methods. *Comput J* 8(1):42. <https://doi.org/10.1093/comjnl/8.1.42>
- Da Silveira G, Borenstein D, Fogliatto F (2001) Mass customization: Literature review and research directions. *Int J Prod Econ* 72(1):1–13. [https://doi.org/10.1016/S0925-5273\(00\)00079-7](https://doi.org/10.1016/S0925-5273(00)00079-7)
- Elgh F (2011) Modeling and management of product knowledge in an engineer-to-order business model. In: ICED 11 - 18th International Conference on Engineering Design—Impacting Society Through Engineering Design, vol 6, pp 86–95
- Elgh F, Cederfeldt M (2007) Concurrent cost estimation as a tool for enhanced producibility-system development and applicability for producibility studies. *Int J Prod Econ* 109(1–2):12–26. <https://doi.org/10.1016/j.ijpe.2006.11.007>
- Elgh F, Johansson J (2014) Knowledge object—a concept for task modelling supporting design automation. In: Moving integrated product development to service clouds in the global economy—Proceedings of the 21st ISPE Inc. International Conference on Concurrent Engineering, CE 2014, pp 192–203. IOS Press BV. https://doi.org/10.3233/978-1-61499-440-4_192
- Fogliatto F, Da Silveira G, Borenstein D (2012) The mass customization decade: an updated review of the literature. *Int J Prod Econ* 138(1):14–25. <https://doi.org/10.1016/j.ijpe.2012.03.002>
- Forrester AI, Keane AJ (2009) Recent advances in surrogate-based optimization. *Prog Aerosp Sci* 45(1–3):50–79. <https://doi.org/10.1016/j.paerosci.2008.11.001>
- Friedenthal S, Moore A, Steiner R (2012) A practical guide to SysML. Elsevier, New York. <https://doi.org/10.1016/B978-0-12-385206-9.10001-8>
- Guin JA (1968) Modification of the complex method of constrained optimization. *Comput J* 10(4):416. <https://doi.org/10.1093/comjnl/10.4.416>
- Hvam L, Mortensen N, Riis J (2008) Product customization. Springer, Berlin. https://doi.org/10.1007/978-3-540-71449-1_1
- Jin R, Chen W, Simpson T (2001) Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13. <https://doi.org/10.1007/s00158-001-0160-4>
- Johansson, J (2007) A flexible design automation system for toolsets for the rotary draw bending of aluminium tubes. In: ASME/IEEE International Conference on Mechatronic and Embedded Systems and Applications and the 19th Reliability, Stress Analysis, and Failure Prevention Conference, vol 4, pp 861–870. ASME. <https://doi.org/10.1115/DETC2007-34310>
- Johansson J (2011) How to build flexible design automation systems for manufacturability analysis of the draw bending of aluminum profiles. *J Manuf Sci Eng* 133(6):061027. <https://doi.org/10.1115/1.4005355>
- Johansson J (2015) Howtomatic suite: a novel tool for flexible design automation. In: Transdisciplinary lifecycle analysis of systems: Proceedings of the 22nd ISPE Inc. International Conference

- on Concurrent Engineering, July 20–23, 2015, no. 2 in *Advances in Transdisciplinary Engineering*, pp 327–336. <https://doi.org/10.3233/978-1-61499-544-9-327>
- Kennedy M, Harmon K, Minnock E (2008) Ready, set, dominate: implement Toyota's set-based learning for developing products and nobody can catch you. Oaklea Press
- La Rocca G, van Tooren M (2012) Knowledge based engineering to support complex product design. *Adv Eng Inf* 26(2):157–158. <https://doi.org/10.1016/j.aei.2012.02.008>
- Pabolu VKR, Stolt R, Johansson J (2016) Manufacturability analysis for welding: a case study using howtomationsuite. In: *Transdisciplinary engineering: crossing boundaries*, no. 4 in *advances in transdisciplinary engineering*, pp 695–704. <https://doi.org/10.3233/978-1-61499-703-0-695>
- Pardalos P, Resende MGC (2002) *Handbook of applied optimization*, vol 26
- Pine B, Davis S (1999) *Mass customization: the new frontier in business competition*. Harvard Business School Press, Harvard
- Poorkiany M, Johansson J, Elgh F (2016) Capturing, structuring and accessing design rationale in integrated product design and manufacturing processes. *Adv Eng Inf* 30(3):522–536. <https://doi.org/10.1016/j.aei.2016.06.004>
- Rudberg M, Wikner J (2004) Mass customization in terms of the customer order decoupling point. *Prod Planning Control* 15(4):445–458. <https://doi.org/10.1080/0953728042000238764>
- Schreiber GT, Akkermans H (2000) *Knowledge engineering and management: the commonKADS methodology*. MIT Press, Cambridge
- Simon H (1973) The structure of ill structured problems. *Artif Intell* 4(3–4):181–201. [https://doi.org/10.1016/0004-3702\(73\)90011-8](https://doi.org/10.1016/0004-3702(73)90011-8)
- Zhu GN, Hu J, Qi J, Ma J, Peng YH (2015) An integrated feature selection and cluster analysis techniques for case-based reasoning. *Eng Appl Artif Intell* 39:14–22. <https://doi.org/10.1016/j.engappai.2014.11.006>

Free-Form Optimization of A Shell Structure with Curvature Constraint



Masatoshi Shimoda and Kenichi Ikeya

Abstract We present a free-form optimization method for designing the optimal shape of a shell structure with curvature constraint. Compliance is minimized under the volume and the state equation constraints. In addition, a target mean curvature of the surface is considered as the equality constraint in order to control the free-form of the shell. It is assumed that a shell is arbitrarily varied in the out-of-plane direction to the surface to create the optimal free-form. A parameter-free, or a node-based shape optimization problem is formulated in a distributed-parameter system based on the variational method. The distribution of the discrete mean curvature is calculated by the area strain obtained from the material derivative formula. The shape gradient function for this problem is theoretically derived using the Lagrange multiplier method and the adjoint variable method, and is applied to the H^1 gradient method for shells. With the proposed method, the optimal free-form of a shell structure with curvature constraint can be efficiently determined. The validity and effectiveness of the method is verified through the numerical examples and the influence of the curvature constraint is demonstrated.

Introduction

In engineering designs, shell structures are abundantly used, since they are thin and light and hold large applied loads effectively. Especially, from view points of

M. Shimoda (✉)

Department of Advanced Science and Technology,
Toyota Technological Institute, 2-12-1 Hisakata, Tenpaku-ku, Nagoya, Japan
e-mail: shimoda@toyota-ti.ac.jp

K. Ikeya

Graduate School of Advanced Science and Technology, Toyota Technological Institute, 2-12-1 Hisakata, Tenpaku-ku, Nagoya, Japan

weight reduction, structural performances, manufacturability and designability, the curvature design of the surface is highly important and the optimization technique become a strong tool for it.

In the previous works concerning the shape optimization of shell structures many parametric methods were presented (Belegundu and Rajan 1988; Ramm et al. 1993; Uysal et al. 2007), where the shell surface was parameterized in advance and the shape parameters are optimized by using an optimization method such as a mathematical programming, an evolutionary method. This is troublesome for designers and an inevitable process in general parametric shape optimization method. In addition, the optimal shape obtained is strongly influenced by the parameterization. On the contrary, papers concerning a non-parametric, or a node-based method are vastly fewer (Leiva 2010). As one of the non-parametric shape optimization methods, we have developed the free-form optimization method (Shimoda and Yang 2014) with the H^1 gradient method (Azegami 1994) for designing a natural and optimal shell form, which is based on the classical variational method and modern numerical optimization method. With this method the optimal shell structure with smooth and arbitrarily free-formed surface can be obtained without any shape parameterization. In our previous works we applied this method to stiffness (Shimoda and Yang 2014), buckling (Shimoda et al. 2016) and natural vibration (Shi et al. 2017) problems. However, the curvature constraint has not been considered in spite of its importance or effectivity. In other methods for shells, there has seldom study the curvature constraint of shell structures. Discrete curvature, or approximate curvature of a discretized surface of Finite Element model has a very important role for the curvature constraint of shell structures. Many methods have been proposed so far to calculate discrete curvature in the research fields of differential geometry, CAD, CG and so on (Eigensatz et al. 2008; Cazals and pouget 2005; Grinspun et al. 2006; Mesmoudi et al. 2010; Zhang et al. 2008). For example, Meyer et al. proposed a discrete method for discrete mean curvature vector by using the Voronoi region area (Meyer et al. 2002). Lui et al. proposed a method using the conformal parameterization (Lui et al. 2008).

In the present work, we consider the mean curvature of the shell surface as the constraint condition for controlling the free-form of the shell. We use the compliance as the objective functional to evaluate the stiffness of the shell. The design variable is a shape variation function distributed on the shell surface, which is arbitrarily varied in the normal direction to the surface. The compliance minimization problem is formulated in a distributed-parameter system based on the variational method. The sensitivity function, or the shape gradient function with respect to the shape variation and the optimal conditions for this problem are theoretically derived using the Lagrange multiplier method, material derivative method (Choi and Kim 2005) and the adjoint variable method (Choi and Kim 2005). The derived shape gradient functions are applied to the H^1 gradient method for shells (Shimoda and Yang 2014). The area strain method (Shimoda and Yang 2014) employed to calculate a discrete mean curvature on the finite element mesh is also introduced. The proposed method is applied to a shell structure with the curvature constraint, and the influence of the difference of the curvature constraint is investigated.

Governing Equation for a Shell

As shown in Fig. 1, we consider a shell having an initial bounded domain $\Omega \subset \mathbb{R}^3$ with the boundary $\partial\Omega$, mid-surface A with the boundary ∂A , side surface S and thickness t . It is assumed for simplicity that a shell structure occupying a bounded domain is a set of infinitesimal flat surfaces, and the Mindlin-Reissner plate theory is employed concerning plate bending. Using the sign convention in Fig. 1, the displacement vector $\mathbf{u} = \{u_i\}_{i=1,2,3}$ of the mid-surface of the shell expressed by the local coordinates is considered by dividing it into the displacements in the in-plane direction $\{u_{0\alpha}\}_{\alpha=1,2}$ and in the out-of-plane direction $u_3 (= w)$. The weak form of the state equation with respect to $(\mathbf{u}_0, w, \boldsymbol{\theta}) \in U$ can be expressed as Eq. (1) (Shimoda and Yang 2014).

$$a((\mathbf{u}_0, w, \boldsymbol{\theta}), (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})) = l((\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})), \forall (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) \in U \tag{1}$$

where $\{\theta_\alpha\}_{\alpha=1,2}$ expresses the rotational angle vector of the mid-surface of the shell. The energy bilinear form $a(\cdot, \cdot)$ and the linear form $l(\cdot)$ for the state variables are respectively defined as:

$$\begin{aligned}
 a((\mathbf{u}_0, w, \boldsymbol{\theta}), (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})) &= \int_{\Omega} \{ E_{\alpha\beta\gamma\delta} (u_{0\alpha,\beta} - x_3\theta_{\alpha,\beta}) (\bar{u}_{0\gamma,\delta} - x_3\bar{\theta}_{\gamma,\delta}) \\
 &\quad + E_{\alpha\beta}^S (w_{,\alpha} - \theta_\alpha) (\bar{w}_{,\beta} - \bar{\theta}_\beta) \} d\Omega \tag{2} \\
 l((\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})) &= \int_{A_d} (f_\alpha \bar{u}_{0\alpha} - m_\alpha \bar{\theta}_\alpha + q \bar{w}) dA \\
 &\quad + \int_A t (b_\alpha \bar{u}_{0\alpha} + b_3 \bar{w}) dA + \int_{\partial A_g} (N_\alpha \bar{u}_{0\alpha} ds - M_\alpha \bar{\theta}_\alpha + Q \bar{w}) ds \tag{3}
 \end{aligned}$$

where $(\bar{\cdot})$ denotes a variation. In this paper, the subscripts of the Greek letters are expressed as $\alpha = 1, 2$, and the tensor subscript notation uses Einstein's summation convention and a partial differential notation for the spatial coordinates $(\cdot)_{,i} = \partial(\cdot)/\partial x_i$. Loads defined as $q, \mathbf{f} = \{f_\alpha\}_{\alpha=1,2}, \mathbf{m} = \{m_\alpha\}_{\alpha=1,2}, \mathbf{N} = \{N_\alpha\}_{\alpha=1,2}, Q, \mathbf{M} = \{M_\alpha\}_{\alpha=1,2}$ and $hb = \{hb_i\}_{i=1,2,3}$ denote an out-of-plane force, an in-

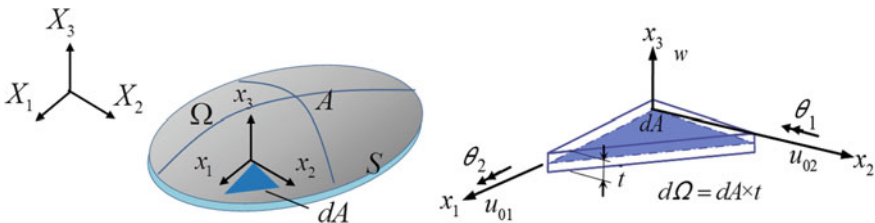


Fig. 1 Shell as a set of infinitesimal flat surface

plane forces, an out-of-plane moments, an in-plane forces, a shearing force, a bending moments and a body force, respectively. In addition, $\{E_{\alpha\beta\gamma\delta}\}_{\alpha,\beta,\gamma,\delta=1,2}$ and $\{E_{\alpha\beta}^S\}_{\alpha,\beta=1,2}$ express an elastic tensor including the bending and the membrane components, and an elastic tensor with respect to the shearing component, respectively. It should be noted that U in Eq. (1) is given by:

$$U = \{(u_{01}, u_{02}, w, \theta_1, \theta_2) \in (H^1(\mathbf{x} \in A))^5 \mid \text{satisfying the given Dirichlet conditions on each subboundary}\} \tag{4}$$

where H^1 is the Sobolev space of order 1.

Free-Form Optimization Problem

Formulation of Free-Form Optimization Problem with Curvature Constraint

In this study, with the aim of maximizing the stiffness of a shell structure, compliance $l(\mathbf{u}_0, w, \boldsymbol{\theta})$ is used as an index of structural stiffness. Letting the curvature, the volume and the state equation in Eq. (1) be the constraint conditions and the compliance the objective functional to be minimized, a distributed-parameter shape optimization problem to determine the optimal shape variation, or the optimal design velocity field \mathbf{V} can be formulated as:

$$\text{Given } A \tag{5}$$

$$\text{Find } A_s \text{ (or } \mathbf{V}) \tag{6}$$

$$\text{that minimizes } l(\mathbf{u}_0, w, \boldsymbol{\theta}) \tag{7}$$

subject to Eq. (1),

$$M \left(\equiv \int_S A \, dS \right) \leq \widehat{M}, \tag{8}$$

and

$$\int_A (|\kappa(\mathbf{x})| - \widehat{\kappa}) \, dA = 0, \quad \mathbf{x} \in A \tag{9}$$

where M and \widehat{M} denote the volume and its constraint value, respectively. κ and $\widehat{\kappa}$ denote the mean curvature and its target value, respectively.

Derivation of Shape Gradient Function

Letting $(\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})$, Λ and Λ_κ denote the Lagrange multipliers for the state equation, the volume constraint and the curvature constraint, respectively, the Lagrange functional L associated with this problem can be expressed as:

$$L = l(\mathbf{u}_0, w, \boldsymbol{\theta}) - a((\mathbf{u}_0, w, \boldsymbol{\theta}), (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})) + l(\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) + \Lambda \left(m - \widehat{M} \right) + \Lambda_\kappa \int_A (|\kappa(\mathbf{x})| - \hat{\kappa}) dA \quad (10)$$

Using the design velocity field \mathbf{V} to represent the amount of domain variation and assuming that the loaded surface or boundary is not varied, the material derivative (Azegami 1994; Choi and Kim 2005) \dot{L} of the Lagrange functional L can be expressed as:

$$\dot{L} = l(\mathbf{u}'_0, w', \boldsymbol{\theta}') - a((\mathbf{u}'_0, w', \boldsymbol{\theta}'), (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})) - a((\mathbf{u}_0, w, \boldsymbol{\theta}), (\bar{\mathbf{u}}'_0, w', \bar{\boldsymbol{\theta}}')) + l(\bar{\mathbf{u}}'_0, \bar{w}', \bar{\boldsymbol{\theta}}') + \Lambda'(M - \widehat{M}) + \Lambda'_\kappa \int_A (|\kappa(\mathbf{x})| - \hat{\kappa}) dA + \langle \mathbf{Gn}, \mathbf{V} \rangle \quad (11)$$

where $(\cdot)'$ expresses the shape derivative (Choi and Kim 2005).

The optimality conditions of the Lagrange functional L with respect to the state variables $(\mathbf{u}_0, w, \boldsymbol{\theta})$, the adjoint variables $(\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})$, Λ and Λ_κ are expressed as shown below.

$$a((\mathbf{u}_0, w, \boldsymbol{\theta}), (\bar{\mathbf{u}}'_0, \bar{w}', \bar{\boldsymbol{\theta}}')) = l((\bar{\mathbf{u}}'_0, \bar{w}', \bar{\boldsymbol{\theta}}')), \forall (\mathbf{u}'_0, w', \boldsymbol{\theta}') \in U \quad (12)$$

$$a((\mathbf{u}'_0, w', \boldsymbol{\theta}'), (\bar{\mathbf{u}}'_0, \bar{w}', \bar{\boldsymbol{\theta}}')) = l(\mathbf{u}'_0, w', \boldsymbol{\theta}'), \forall (\mathbf{u}'_0, w', \boldsymbol{\theta}') \in U \quad (13)$$

$$\Lambda(M - \widehat{M}) = 0, M - \widehat{M} \leq 0, \Lambda \geq 0 \quad (14)$$

$$\int_A (|\kappa(\mathbf{x})| - \kappa_a) dA = 0 \quad (15)$$

The state equation (Eq. 12) for $(\mathbf{u}_0, w, \boldsymbol{\theta})$ and the adjoint equation (Eq. 13) for $(\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})$ can be solved with a standard commercial FEA code. By considering the self-adjoint relationships $(\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) = (\mathbf{u}_0, w, \boldsymbol{\theta})$ obtained from Eqs. (12) and (13) and by substituting $(\mathbf{u}_0, w, \boldsymbol{\theta})$ and Λ and Λ_κ determined by these optimality conditions into Eq. (11), the material derivative \dot{L} can be expressed as the dot product of the shape gradient function \mathbf{Gn} ($=\mathbf{G}$) and the design velocity field \mathbf{V} as shown in the following equation.

$$\dot{L} = \langle \mathbf{Gn}, \mathbf{V} \rangle \quad (16)$$

$$\langle \mathbf{Gn}, \mathbf{V} \rangle = \int_A G_A V_n dA + \int_A G_C V_n dA \quad (17)$$

where the shape gradient density functions (or the shape sensitivity function) G_A and G_C for this problem are derived as

$$G_A = - \left\{ C_{\alpha\beta\gamma\delta} \left(u_{0\alpha,\beta} - \frac{t}{2} \theta_{\alpha,\beta} \right) \left(\bar{u}_{0\gamma,\delta} - \frac{t}{2} \bar{\theta}_{\gamma,\delta} \right) - C_{\alpha\beta\gamma\delta} \left(u_{0\alpha,\beta} + \frac{t}{2} \theta_{\alpha,\beta} \right) \left(\bar{u}_{0\gamma,\delta} + \frac{t}{2} \bar{\theta}_{\gamma,\delta} \right) \right\} + 2\Lambda t \kappa(x) \quad (18)$$

$$G_C = \Lambda_\kappa \left\{ \frac{\partial |\kappa(\mathbf{x})|}{\partial n} + 2(|\kappa(\mathbf{x})| - \hat{\kappa}) \kappa(\mathbf{x}) \right\} \quad (19)$$

The first and second terms of G_A are the shape gradient density functions for the compliance and the volume constraint, respectively. The shape gradient density function G_C is for the curvature constraint.

Calculation of Discrete Mean Curvature

The shape gradient density functions include the mean curvature κ of the shell surface. When we calculate Eq. (12) by FEM, we have to approximately calculate the mean curvature at all nodes. In several methods for the approximate curvature of a discretized surface, or a discrete curvature, we employ the method based on the area strain (called area strain method) by Shimoda and Yang (2014), since it has the common theoretical background (or the material derivative method). In the method employed, the discrete mean curvature κ on the finite element model is calculated by the area strain of a small area around the node by the small variation in the normal direction to the elements, which is based on the material derivative formula.

For a given functional J and a distributed function ψ shown in Eq. (20), the material derivative formula is expressed by Eq. (21) (Shimoda and Yang 2014; Choi and Kim 2005).

$$J = \int_A \psi dA \quad (20)$$

$$\dot{J} = \int_A \{ \psi' + (\psi_{,i} n_i + 2\psi \kappa) V_n \} dA \quad (21)$$

When the distributed function $\psi = 1$, J expresses a surface area A . The material derivative \dot{A} becomes

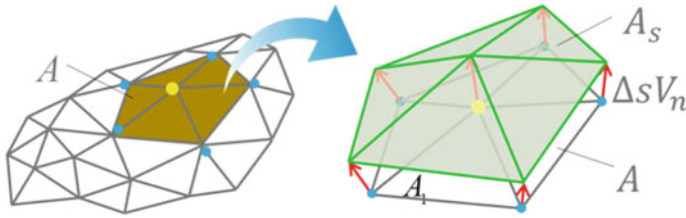


Fig. 2 Schematic of calculation of discrete mean curvature using the area strain

$$\dot{A} = 2 \int_{\Gamma} \kappa V_n d\Gamma = 2\kappa V_n = \frac{\Delta A}{\Delta s} \tag{22}$$

Then,

$$\kappa = \frac{\Delta A}{2\Delta s V_n A} \tag{23}$$

By giving a small variation in the normal direction to the surface $\Delta s V_n$, κ can be calculated. $\Delta A/A$ expresses the area strain. Applying Eq. (23) to FE model as shown in Fig. 2, the discrete mean curvature can be calculated.

In order to validate the area strain method employed, discrete mean curvatures were evaluated for a discretized sphere with a radius $R = 1$ as shown in Fig. 3. Table 1 shows the comparison of accuracy with Meyer’s method (Meyer et al. 2002), where all nodes are evaluated. The mean and maximum absolute errors to the theoretical values of the employed method are only 0.03 and 0.01%, respectively. Meyer’s method also shows good agreement.



Fig. 3 Discretized sphere model for evaluation of discrete mean curvatures

Table 1 Comparison of accuracy of discrete curvature calculations

Mean absolute error		Max. absolute error	
Mayer	Employed	Mayer	Employed
1.32×10^{-4}	3.18×10^{-4}	1.32×10^{-4}	1.32×10^{-4}

Calculation of Shape Gradient Density Function

The shape gradient density function G_A of Eq. (18) can be easily calculated by the stress and strain on the top and the bottom surface of the shell. Calculating $\partial|\kappa(\mathbf{x})|/\partial n$ in G_C of Eq. (19), we assume that local small surface is approximated by the sphere function. Then, $\partial|\kappa(\mathbf{x})|/\partial n$ is expressed as Eq. (24).

$$\frac{d|\kappa|}{dn} = \begin{cases} \kappa^2 & (\kappa \geq 0) \\ -\kappa^2 & (\kappa < 0) \end{cases} \tag{24}$$

where the following relationships are introduced.

$$\kappa \simeq \frac{1}{2} \left(\frac{1}{r} + \frac{1}{r} \right), \quad \frac{d\kappa}{dr} = -\kappa^2 \tag{25}$$

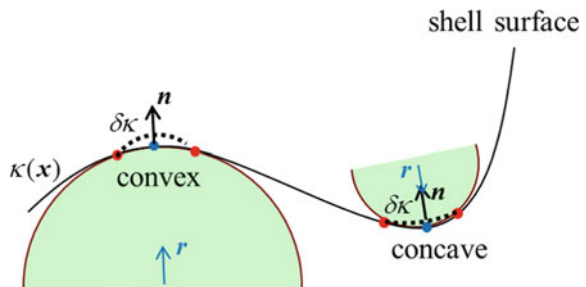
$$\mathbf{n} = \begin{cases} \mathbf{r} & (\kappa \geq 0) \\ -\mathbf{r} & (\kappa < 0) \end{cases} \tag{26}$$

where r and \mathbf{r} are the radius and the unit radius vector of the approximated local surface, respectively as shown in Fig. 4. \mathbf{n} is the unit normal vector to the shell surface.

Using these relationships, the shape gradient density function G_C for the curvature constraint is simply expressed as

$$G_C = \begin{cases} \kappa^2 + (\kappa - \hat{\kappa})\kappa = 2\kappa^2 - \hat{\kappa}\kappa & (\kappa \geq 0) \\ -\kappa^2 + (-\kappa - \hat{\kappa})\kappa = -(2\kappa^2 + \hat{\kappa}\kappa) & (\kappa < 0) \end{cases} \tag{27}$$

Fig. 4 Schematic of curvature variation for convex or concave shell surface



The shape gradient density functions calculated are applied to the H^1 gradient method for shells to determine the optimal design velocity field \mathbf{V} .

H^1 Gradient Method for Shells

The free-form optimization method for shells consists of main three processes; (1) Derivation of shape gradient function, (2) Numerical calculation of shape gradient function, (3) The H^1 gradient method for determining the optimal shape variation, or the optimal velocity field. The H^1 gradient method is a gradient method in a Hilbert space, which was originally proposed by Azegami in 1994. Shimoda et al. extended it for designing free-form shells (Shimoda and Yang 2014). It is a non-parametric or a node-based shape optimization method that can treat all nodes as design variables and does not require any design parameterization.

The H^1 gradient method for shells is briefly introduced as follows. When the optimality conditions of Eqs. (12)–(15) are satisfied, the perturbation expansion of the Lagrange functional L can be written as

$$\Delta L = \langle \mathbf{G}\mathbf{n}, \Delta s(\mathbf{V}, \boldsymbol{\theta}) \rangle + O(|\Delta s|^2) \quad (28)$$

where Δs is a small positive value.

Considering the design velocities $\mathbf{V} = \{V_i\}_{i=1,2,3}$ as a combination of the in-plane velocities $\{V_{0\alpha}\}_{\alpha=1,2}$ and the out-of-plane velocity V_3 , the following equation is introduced as the governing equation for $\mathbf{V} = (V_{01}, V_{02}, V_3)$.

$$\begin{aligned} & \alpha \langle (\mathbf{V}_{0\alpha}, V_3, \boldsymbol{\theta}), (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) \rangle + \alpha \langle (\mathbf{V} \cdot \mathbf{n})\mathbf{n}, (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) \rangle \\ & = -\langle \mathbf{G}\mathbf{n}, (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) \rangle, (\mathbf{V}_{0\alpha}, V_3, \boldsymbol{\theta}) \in C_{\ominus}, \forall (\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}}) \in C_{\ominus} \end{aligned} \quad (29)$$

$$\begin{aligned} C_{\ominus} = & \left\{ (V_1, V_2, V_3, \theta_1, \theta_2) \in (H^1(S))^5 \mid \text{satisfying the given} \right. \\ & \left. \text{Dirichlet condition for shape variation on } \partial \mathbf{A} \right\} \end{aligned} \quad (30)$$

where α is a positive constant, C_{\ominus} is the kinematically admissible function space.

Considering the arbitrariness of $(\bar{\mathbf{u}}_0, \bar{w}, \bar{\boldsymbol{\theta}})$ in Eq. (29), we substitute Eq. (29) into Eq. (28), and obtain

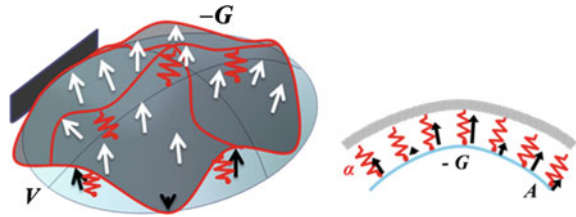
$$\Delta L \cong \langle \mathbf{G}\mathbf{n}, \Delta s(\mathbf{V}, \boldsymbol{\theta}) \rangle = -\{a_L((\mathbf{V}, \boldsymbol{\theta}), \Delta s(\mathbf{V}, \boldsymbol{\theta})) + \alpha \langle (\mathbf{V} \cdot \mathbf{n})\mathbf{n}, \Delta s(\mathbf{V}, \boldsymbol{\theta}) \rangle\} \quad (31)$$

Taking into account the positive-definiteness of the bilinear form on the left side of Eq. (29), which is

$$\alpha \langle (\mathbf{V}, \boldsymbol{\theta}), \Delta s(\mathbf{V}, \boldsymbol{\theta}) \rangle + \alpha \langle (\mathbf{V} \cdot \mathbf{n})\mathbf{n}, \Delta s(\mathbf{V}, \boldsymbol{\theta}) \rangle > 0 \quad (32)$$

and $\alpha > 0$, and $\Delta s > 0$, the following relationship is obtained:

Fig. 5 Schematic of the H^1 gradient method for shells



$$\Delta L < 0 \quad (33)$$

This relationship guarantees that the Lagrange functional is definitely reduced in a piecewise convex design space by updating the form of shell by the design velocity field V determined from Eq. (29).

Here, let us consider this gradient method based on Eq. (29) from the point of view of structural analysis. In this method, the negative shape gradient function $-Gn (= -G)$ is applied as a distributed force to a fictitious-linear elastic shell in the normal direction to the design surface under a Robin boundary condition, i.e., an elastic support condition with a distributed spring constant α . The shape gradient function is not used directly while replaced by a distributed force to vary the shape. α is employed to control the influence range of the shape gradient function, and to avoid rigid motion of the shape variation. This approach makes it possible both to reduce the objective function and to maintain surface smoothness, i.e., mesh regularity (Azegami et al. 1997), simultaneously, which is the most distinctive feature of the H^1 gradient method for shells. The design velocity field V is calculated as the displacement field obtained from this linear elastic analysis (called velocity analysis) and is used to update the shape. The shape design constraint condition is arbitrarily set by the boundary conditions in the velocity analysis. The H^1 gradient method for shells is schematized in Fig. 5. This analysis can be also solved with a standard commercial FEM code. We use MSC/NASTRAN in this research.

The optimization process of the developed system is schematized in Fig. 6. We have developed our optimization system on Windows platform. The C programs on it have been developed using the C compiler. The C programs and FEM commercial code are integrated and controlled by the batch commands of Windows OS. In the optimization process, firstly, the stiffness analysis is done by a commercial FEM code and the outputs of the analysis are utilized to calculate the sensitivity of Eq. (18). After that, the velocity analysis with compliance sensitivity (or the first term of G_A) is implemented to determine the design velocity field V , and the shape is updated using V . Next, the curvature sensitivity of G_C is calculated. The velocity analysis is implemented with G_C , and the shape is re-updated. Then, the curvature constraint is judged and repeated if it is not satisfied. Next (option), the volume constraint is judged. If it is not satisfied, the volume sensitivity (or the second term of G_A) is calculated. The velocity analysis is implemented with it, and the shape is re-updated. Then, the curvature constraint is re-judged. This process is repeated until the objective function is converged.

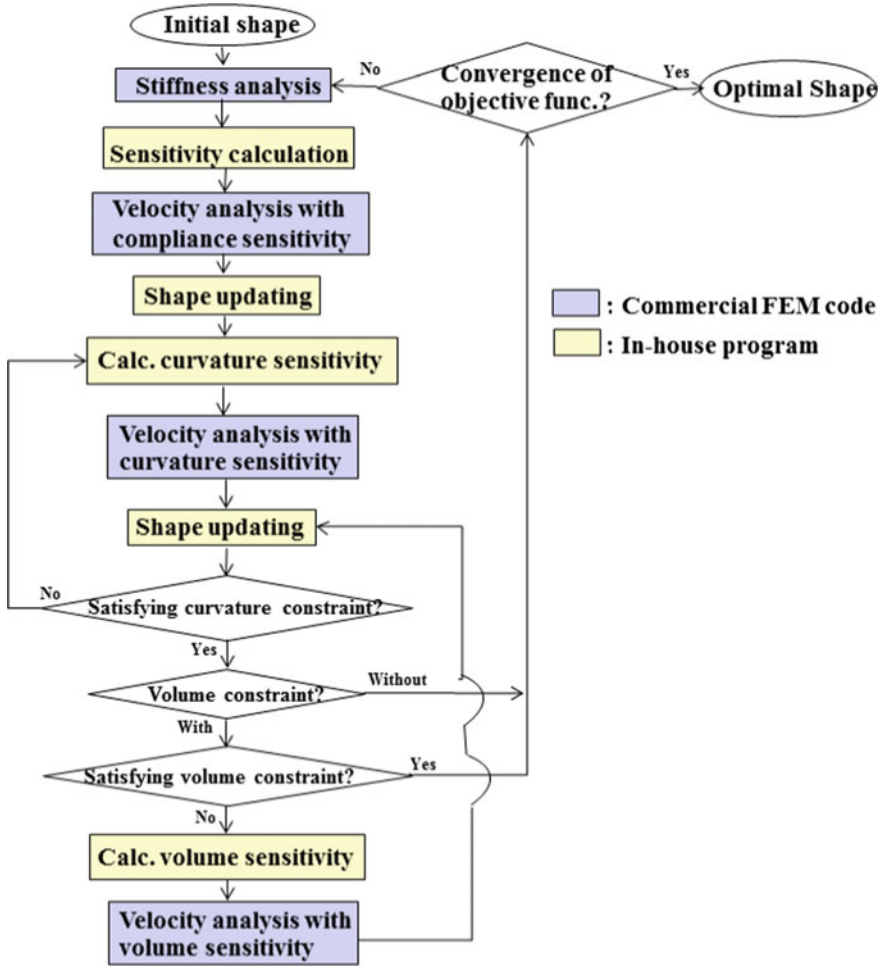


Fig. 6 Free-form optimization process of the developed system

Calculated Results

The proposed method was applied to T-joint model. The initial shape and the problem definition are illustrated in Fig. 7. In the stiffness analysis shown in Fig. 7a, the left and right side edges of T-joint were simply supported. A distributed force was applied at the top edges. In the velocity analysis shown in Fig. 7b, it was assumed that the right and left side edges, the top and bottom surfaces were simply supported. The volume constraint was set as 1.0 times the initial value. The material constants were used as $E = 210000$ Pa and $\nu = 0.3$. The obtained shape with the curvature constraint is shown in Fig. 8. The target curvature value $\hat{\kappa}$ was set as 0.2

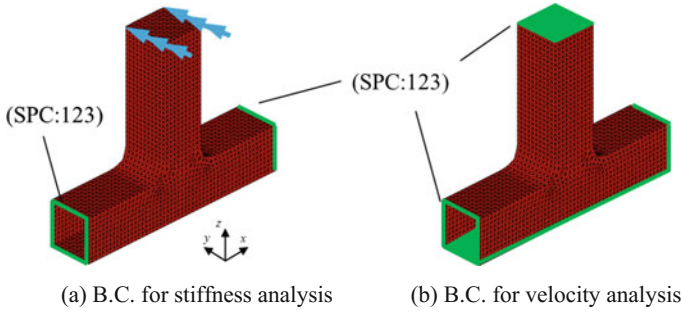
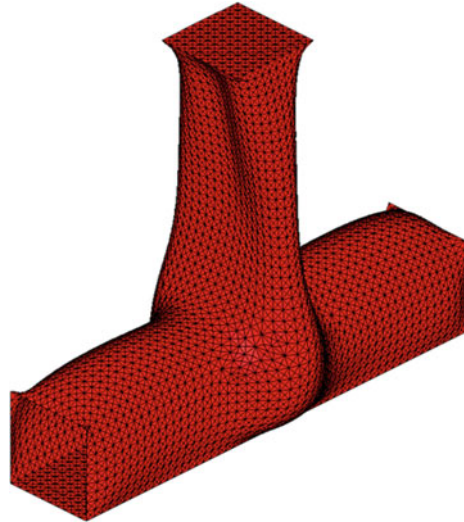


Fig. 7 Initial structure and boundary conditions of T-joint model

Fig. 8 Obtained shape of T-joint model with curvature constraint (isotropic material $\hat{\kappa} = 0.2 * \text{Max. mean curvature of the initial shape}$)



times the maximum mean curvature of the initial shape. The bottom of the pillar was expanded and the upper part was shrunk while maintaining the surface smoothness. Figure 9 shows the iteration convergence history of the volume and the compliance. The compliance decreased by around 56.5% at 47 iterations and converged steadily while satisfying the volume constraint. The curvature constraint was also satisfied actively, which was omitted in the graph because all values were zero. For comparison, the obtained shape without the curvature constraint is shown in Fig. 10a and superposed in b. The obtained shape has large curvatures by the free variation for minimizing the compliance, and is clearly different from Fig. 8. Figure 11 shows the iteration convergence history without the curvature constraint. The compliance decreased by around 62%. Figure 12 shows the obtained shape with another curvature constraint, of which the target curvature value $\hat{\kappa}$ was set as 0.1 times the maximum mean curvature of the initial shape. The curvature distribution was suppressed small while maintaining the surface smoothness. In this case, the compliance

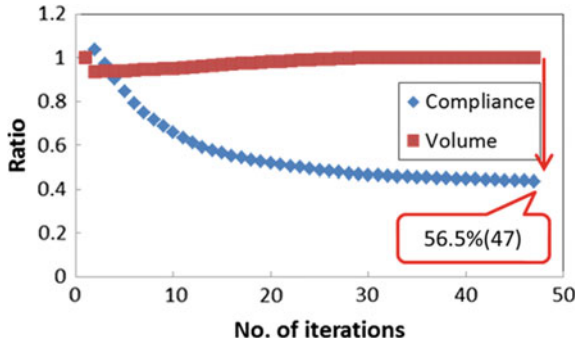
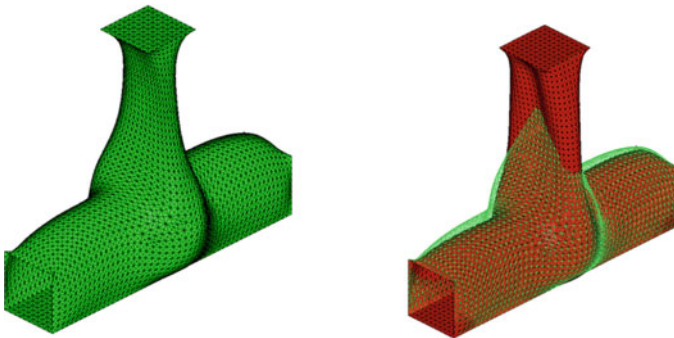


Fig. 9 Iteration histories of T-joint model with curvature constraint ($\hat{\kappa} = 0.2 * \text{Max. mean curvature of the initial shape}$)



(a) Optimal shape (without curvature const.) (b) Superposing (red: with curvature const., green: without curvature const.)

Fig. 10 Comparison of optimal shapes ($\hat{\kappa} = 0.2 * \text{Max. mean curvature of the initial shape}$)

and the volume respectively decreased by around 20 and 10% because of the more strict curvature constraint. It was confirmed that the compliance reduction become smaller by imposing the curvature constraint, as was expected. From these results, validity of the proposed method was verified.

Conclusion

This paper proposed a non-parametric free-form optimization method of shell structures with curvature constraint. The shape gradient function was derived using the material derivative method and the adjoint variable method. The discrete mean curvature was evaluated by the area strain based on the material derivative formula. Design examples were demonstrated to verify the effectiveness and practical utility

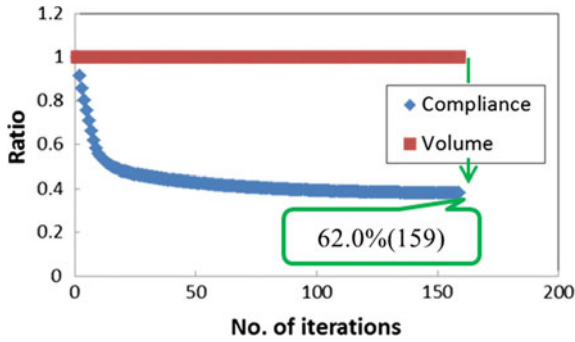
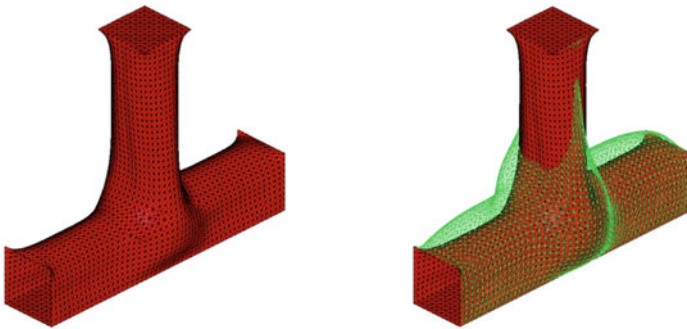


Fig. 11 Iteration histories of T-joint model without curvature constraint



(a) Optimal shape (without curvature const.) (b) Superposing (red: with curvature const., green: without curvature const.)

Fig. 12 Comparison of optimal shapes ($=0.1 \cdot \text{Max. mean curvature of the initial shape}$)

of this method. The proposed method makes it possible to obtain the smooth and natural optimal form without shape design parameterization, while satisfying the volume and curvature constraints and reducing the objective functional.

In our future work, we will apply the proposed method to the practical design problems of car bodies, architectures and other problems with aesthetic evaluation function. Also, we will combine the curvature constraint method with other objective functions such as maximum stress, buckling load, vibration eigenvalue and so on.

Acknowledgements This work was supported by a grant-in-aid from the Research Center for Smart Vehicles at Toyota Technological Institute.

References

- Azegami H (1994) A solution to domain optimization Problems. *Trans Jpn Soc Mech Eng, Ser A* 60:1479–1486 (in Japanese)
- Azegami H, Kaizu S, Shimoda M, Katamine E (1997) Irregularity of shape optimization problems and an improvement technique. *Comput Aided Optimum Des Struct V*: 309–326
- Belegundu AD, Rajan SD (1988) A shape optimization approach based on natural design variables and shape functions. *Comput Methods Appl Mech Eng* 66:87–106
- Cazals F, pouget M (2005) Estimating differential quantities using polynomial fitting of osculating jets. *Comput Aided Geometrical Des* 22(2):121–146
- Choi KK, Kim NH (2005) *Structural sensitivity analysis and optimization 1*. Springer, New York
- Eigensatz M, Sumner RW, Pauly M (2008) Curvature-domain shape processing. *Eurographics* 27(2):2008
- Grinspun E, Gingold Y, Reisman J, Zorin D (2006) Computing discrete shape operators on general meshes. *Comput Graph Forum* 25:3
- Leiva PL (2010) Freeform optimization: a new capability to perform grid by grid shape optimization of structures. In: *Proceedings of 6th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems*, Kyoto
- Lui LM, Kwan J, Wang Y, Yau ST (2008) Computation of curvatures using conformal parameterization. *Commun Inf Syst* 8(1):1–16
- Mesmoudi MM, De Florian L, Magillo P (2010) A geometric approach to curvature estimation on triangulated 3D shapes. In: *International Conference on Computer Graphics Theory and Applications*, 17–21 May 2010. pp 90–95
- Meyer M, Desbrun M, Schroder P, Barr AH (2002) Discrete differential-geometry operators for triangulated 2-manifolds. *Vis Math III*:35–57
- Ramm UE, Bletzinger KU, Reiting R (1993) Shape optimization of shell structures. *J Int Assoc Shell Spat Struct* 34(2):103–121
- Shi JX, Nagano T, Shimoda M (2017) Fundamental frequency maximization of orthotropic shells using a free-form optimization method. *Compos Struct* 170:135–145
- Shimoda M, Yang L (2014) A non-parametric free-form optimization method for shell structures. *Struct Multidisc Optim* 50:409–423
- Shimoda M, Okada T, Nagano T, Shi JX (2016) Free-form optimization method for buckling of shell structures under out-of-plane and in-plane shape variations. *Struct Multidisc Optim* 54:275–288
- Uysal M, Gul R, Uzman U (2007) Optimum shape design of shell structures. *Eng Struct* 29:80–87
- Zhang X, Li H, Cheng Z (2008) Curvature estimation of 3D point cloud surfaces through the fitting of normal section curvatures. In: *Proceedings of Asia Graph 2008, Japan*, 22–26 Oct 2008. pp 72–79

Application of Game Theory and Evolutionary Algorithm to the Regional Turboprop Aircraft Wing Optimization



Pierluigi Della Vecchia, Luca Stingo, Fabrizio Nicolosi,
Agostino De Marco, Elia Daniele and Egidio D'Amato

Abstract Nash equilibrium and evolutionary algorithm are used to optimize a wing of a regional turboprop aircraft, with the aim to compare different optimization strategies in the aircraft design field. Since the aircraft design field is very complex in terms of number of involved variables and space of analysis, it is not possible to perform an optimization process accounting for all possible parameters. This leads to the need to reduce the number of the variables to the most significant ones. A multi-objective optimization approach is here performed, paying attention to the variables which mainly influence the objective functions. Results of Nash-Genetic algorithm are compared against those of both a typical Pareto front and a scalarization, showing that the proposed approach locates almost all solutions on the Pareto front, while the scalarization results are confined only in a zone of this front. The optimization elapsed time for a single optimization point is less than 32% of an entire Pareto front, but the designer must initially choose the players' cards assignment.

Introduction

Nowadays multi-objective optimization problems are usually solved via Pareto Genetic Algorithms (GAs), to find a wide range of solutions for a given problem, distributing the solutions over the entire Pareto front. A comprehensive overview of GA applied in multi-objective optimization can be found in Fonseca and Fleming (1995).

P. Della Vecchia (✉) · L. Stingo · F. Nicolosi · A. De Marco
Aerospace Division, Department of Industrial Engineering,
University of Naples Federico II, 80125 Naples, Italy
e-mail: pierluigi.dellavecchia@unina.it

E. Daniele
Fraunhofer IWES, K pkersweg 70, 26129 Oldenburg, Germany
e-mail: elia.daniele@iwes.fraunhofer.de

E. D'Amato
Universit  degli Studi della Campania "Luigi Vanvitelli",
Via Roma 29, 81031 Aversa, Italy
e-mail: egidio.damato@unina2.it

  Springer International Publishing AG, part of Springer Nature 2019
E. Andr s-P rez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*, Computational Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_26

Classical GA approaches are cooperative, based on Pareto ranking, using both sharing and mating restrictions to ensure diversity. An alternative approach to solve multi-objective problem is based on John Nash theory (Nash 1950, 1951), this time non-cooperative one, where players aim at solving multiple objective optimization problems originating from Game Theory and Economics. According to this theory, for a given optimization problem with G objectives, a Nash strategy consists in having G players, each of them wants to improve his benefit, based on his criterion, with all the other criteria fixed by the other players. When no player can further improve his benefit, it means that the system has reached a point of equilibrium, called Nash Equilibrium. In the present paper, the Nash strategy is coupled to genetic algorithms, to reduce the computational time and allow a more physic-based association among variables and objective functions.

The Nash GAs (NGA) was introduced by Sefrioui and Periaux (2000), who showed a GA based on the concept of a non-cooperative multiple objective algorithm. They brought together genetic algorithms and Nash strategy to make the GA builds the Nash Equilibrium. In their work, they tested algorithm on different analytic functions and on nozzle optimization, comparing results with GA and single objective weighted strategy, highlighting that NGA was the most robust, less time-consuming approach as it always converges towards a point of the global. Due to its valuable performance, NGA algorithm has been extensively employed in recent years on several fields of engineering and science, mainly, in aeronautical and structural engineering. Particularly, the use of NGA has been demonstrated as acceleration tool increasing convergence speed and/or quality of solutions in computational mechanics application (Periaux et al. 2015). Recently, NGA has been also applied in single objective reconstruction inverse design problem in structural engineering, with successful speed-up of structural optimization search (Greiner et al. 2016), showing an advantage greater when the problem size increases.

Mallozzi et al. (2016) presented a wing two-objective optimization problem is approached by a cooperative model, just minimizing the sum of the weight and the drag, as well by a non-cooperative model by means of the Nash equilibrium concept.

The authors have been also implemented and tested their own NGA algorithm [as shown in D'Amato et al. (2012a, b)]. This algorithm has been successfully applied to multi-objective airfoil optimization (Della Vecchia et al. 2014) and embedded into the Multidisciplinary Design Optimization framework of the AGILE European project (Della Vecchia et al. 2017; Lefebvre et al. 2017). The present paper aims to provide a comparison of NGA with Pareto GA and a single objective weighted function, applied to regional turboprop aircraft wing optimization. The structure of the work is as follows: Section “[The NGA Algorithm](#)” describes the NGA implemented algorithm, Section “[Applications: Wing Optimization](#)” shows two different applications with 2 and 3 players (objective functions) respectively. Finally, conclusions are addressed.

The NGA Algorithm

The regional turboprop aircraft wing optimization problem is here approached by means of game theory solutions, in particular, the Nash equilibrium solution concept, for which no player has anything to gain by unilaterally changing his strategy (Basar and Olsder 1995). Reducing the general multi-player formulation to a two-player situation, the mathematical expression for the Nash equilibrium problem N is:

$$\left\{ \begin{array}{l} \text{find } (\bar{x}_1, \bar{x}_2) \in X_1 \times X_2 \text{ such that} \\ f_1(\bar{x}_1, \bar{x}_2) = \min_{x_1 \in X_1} f_1(x_1, \bar{x}_2), \\ f_2(\bar{x}_1, \bar{x}_2) = \min_{x_2 \in X_2} f_2(\bar{x}_1, x_2) \end{array} \right. \quad (1)$$

where $(x_1, x_2) \in X_1 \times X_2$ are the players' variables or strategies, defined in their own strategy domains X_1, X_2 , while f_1, f_2 are the players' objective functions. In the specific case object of this research, the players' variables are in their self a variables' set, as $x_1 = [\xi_1, \dots, \xi_n]$, $x_2 = [\eta_1, \dots, \eta_m]$ of dimension n, m depending on the variables partition introduced by the optimization problem decomposition, the latter being case specific. The genetic algorithm (GA) is an adaptive heuristic search method based on the principles of genetics and natural selection. Its name sets the roots in the analogy with living organisms in nature, being a GA capable of driving the evolution of a population (in conjunction with game theory, of players) under specified selection rules aiming to maximize their fitness w.r.t. the environment (i.e. an objective function under operating conditions and constraints). A GA structure could be regarded as a composition of the following pieces: (i) a finite set of n -dimensional array, i.e. the population or players, usually encoded as a string of bits named genotype; (ii) an adaptive function, called fitness, that estimates the goodness of the solution, indicating the individuals to let reproduce; (iii) semi random genetic operators such as selection, crossover and mutation that operate on individuals, changing their associated fitness. The constraints are implemented by means of penalty functions, decreasing the individuals' fitness. The solution quality, enhanced by a large population, is also the bane of a GA in simple problems (Haupt and Haupt 2004), leading in general to higher computational time. However, its wide usage is justified by several advantages, among which:

- The use of continuous or discrete variables.
- The trend of the objective function and its derivatives can be unknown.
- It deals with problems with many variables.
- It offers an intrinsic parallelization of the algorithm.
- It delivers satisfactory results in problems with extremely complex objective functions hypersurfaces (i.e. with many local minima).
- It performs properly with numerically and/or experimentally generated data.

These features favor the GA in cases where the traditional optimization approaches fail. The algorithm for a two player Nash equilibrium game (D’Amato et al. 2012a, b) is here described for simplicity.

Let U, V be players’ strategy sets (both are metric spaces). Let f_1, f_2 be two real valued functions defined on $U \times V$ representing the players’ objective functions. The used algorithm is based on the Nash adjustment process (Fudenberg and Tirole 1991), where players take turns setting their outputs, and the chosen output of each player (in U) is his best response to the output previously chosen by his opponents (in V). The converged steady state value of this process is a Nash equilibrium of the game. Let $s = u, v$ be the pair representing the potential solution for a 2 person Nash problem. Then u denotes the subset of variables handled by the player 1, belonging to U , and optimized under the objective function f_1 . Similarly, v indicates the subset of variables handled by the player 2, belonging to V , and optimized under a different objective function f_2 . As stated in the Nash equilibrium definition (Nash 1951), the player 1 optimizes pair s with respect to f_1 by modifying u while v is fixed by the player 2; symmetrically, the player 2 optimizes pair s w.r.t. the f_2 by modifying v , while u is fixed by the player 1. This procedure can be implemented numerically considering u^{k-1} and v^{k-1} be the best values found by players 1 and 2, respectively, at step (or generation) $k - 1$. At next step, k , the player 1 optimizes u^k using v^{k-1} to evaluate the pair $s = u^k, v^{k-1}$. At the same time, the player 2 optimizes v^k using u^{k-1} to evaluate the pair $s = u^{k-1}, v^k$. The algorithm is structured in several phases, see also Fig. 1:

1. Generation of two different random populations, one for each player, at the first step. Player 1’s optimization task is performed by acting on the first population and *vice versa*.
2. The sorting of the individuals among their respective population, is based on the evaluation of a fitness function typical of GAs. The results of the matches between each individual of population 1 with all individuals of population 2 (scoring 1 or -1 , respectively, for a win or loss, and 0 for a draw) are stored, see Eq. (2).

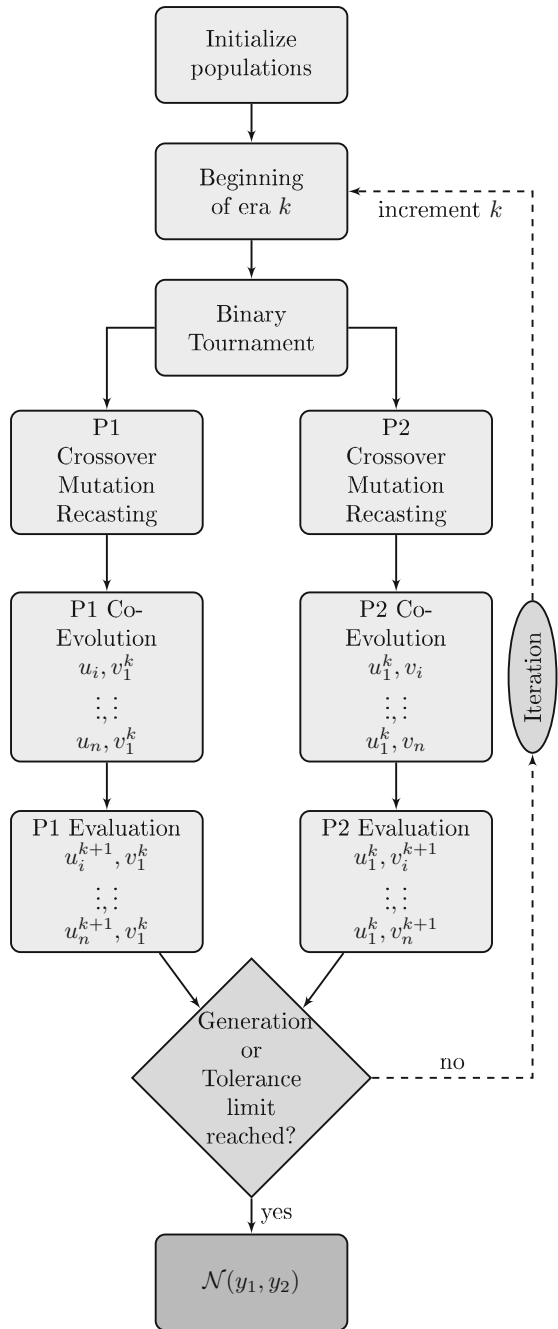
$$\begin{cases} \text{if } f_1(u_i^k, v^{k-1}) > f_1(u^{k-1}, v_i^k), \text{ fitness}_1 = 1 \\ \text{if } f_1(u_i^k, v^{k-1}) < f_1(u^{k-1}, v_i^k), \text{ fitness}_1 = -1 \\ \text{if } f_1(u_i^k, v^{k-1}) = f_1(u^{k-1}, v_i^k), \text{ fitness}_1 = 0 \end{cases} \quad (2)$$

A similar procedure is need for the player 2, as expressed in Eq. (3).

$$\begin{cases} \text{if } f_2(u_i^k, v^{k-1}) > f_2(u^{k-1}, v_i^k), \text{ fitness}_2 = 1 \\ \text{if } f_2(u_i^k, v^{k-1}) < f_2(u^{k-1}, v_i^k), \text{ fitness}_2 = -1 \\ \text{if } f_2(u_i^k, v^{k-1}) = f_2(u^{k-1}, v_i^k), \text{ fitness}_2 = 0 \end{cases} \quad (3)$$

The individuals having an equal fitness value are sorted by f_1 for player 1 and f_2 for player 2.

Fig. 1 In this figure the Nash Genetic Algorithm structure is shown. The sequence is composed by five steps, from the generation of random populations, one for each player, to achieving the Nash equilibrium, through the evaluation of a fitness function based on GAs approach and mutation operations among the individuals of each population. The detailed description is presented in the text



3. A mating pool for parent individuals is established, and crossover and mutation operations are performed on each player population. This new, evolved, population is sorted again, as described in phase 2.
4. At the end of the $k - th$ step, the player 1 delivers his best value, u^k , to player 2 who will use it at step $k + 1$ to assign a unique value for the first part of his pair, i.e. the one depending on player 1, while the second part is that derived from crossover and mutation operations. Conversely, player 2 delivers his best value, v^k , to player 1 who will use it at step $k + 1$, assigning a unique value for the second part of the pair, i.e. the one depending on player 2.
5. A Nash equilibrium is found when a maximum number of steps is reached, by repeating the phases 2-4. This algorithmic structure is similar to some of those used in literatures (Deb et al. 2000), with a major emphasis on fitness function consistency (Wang et al. 2002; Della Vecchia et al. 2014).

Applications: Wing Optimization

This section shows two different applications of NGA algorithm with two players (see Section “Two Players – C_{Dw} Versus W_w ”) and three players (see Section “Three Players – C_{Dw} Versus W_w Versus C_{Lmax} ”) involved, respectively. The idea to apply NGA equilibrium solutions to aircraft design applications has a main driver. It avoids a more arbitrary and less physically based variables association among the different objective functions, using instead a more engineering reliable variables assignment based on well-known parameter association.

Two Players – C_{Dw} Versus W_w

The first application deals with a two players NGA, Γ , to a turboprop wing optimization [see Eq. (4)]. The objective functions or pay-offs are: the wing drag coefficient, C_{Dw} , computed with simple equivalent flat plate method and parabolic drag approximation; the wing weight, W_w , according to the methodology proposed by Raymer (2002). Five design variables have been used in this application, as shown in Eq. (4), including the wing aspect ratio AR, the mean wing thickness t/c , the wing area S_w , the leading edge swept back angle Δ_{LE} , and the taper ratio λ . These variables are assigned among the two players in all possible combinations, leading to 30 games among the players.

$$\begin{aligned}
\Gamma &= \langle \text{players} : 2; \\
AR &: \{11.45 - 13.26\}, \\
\frac{t}{c} &: \{0.14 - 0.18\}, \\
S_w &: \{55.27 - 70.1\}, \\
\Lambda_{LE} &: \{0 - 3\}, \\
\lambda &: \{0.45 - 0.64\}; \\
&C_{Dw}, W_w \rangle.
\end{aligned} \tag{4}$$

In order to consider the effect on the overall aircraft weight the following considerations have been done: (I) the aircraft weight is calculated as shown in Eq. (5), summing up the operative empty weight W_{OE} , the payload $W_{Payload}$ and the fuel W_{Fuel} ; (II) the wing weight, which affects the overall aircraft weight, is evaluated thanks to Eq. (8), where W_{dg} and N_Z represent the design gross weight and the ultimate load factor respectively; (III) W_{OE} and W_{Fuel} are calculated according to Eqs. (6) and (7), respectively, where W_{OE_ref} and $W_{wing_initial}$ are the initial reference weights; (IV) based on the aircraft cruise lift coefficient C_L , computed for each wing during the optimization process as the cruise lift coefficient, the aircraft drag coefficient is computed through Eq. (9), where the AR and the Oswald factor e vary for each wing. Equations (10) and (11) represent, respectively, the objective functions: F_{obj_1} considers the Prandtl-Glauert compressibility correction (M_{corr}); F_{obj_2} is the non-dimensional weight objective function.

$$W_{AC} = W_{payload} + W_{OE} + W_{Fuel}, \tag{5}$$

$$W_{OE} = W_{OE_ref} + F_{obj_2} \cdot W_{wing_initial}, \tag{6}$$

$$W_{Fuel} = 0.54 \cdot \frac{S_w^2}{b} \cdot \left(\frac{t}{c}\right) \cdot \frac{1 + \lambda \cdot \sqrt{\frac{5}{6} \cdot \left(\frac{t}{c}\right) + \lambda^2 \cdot \left(\frac{t}{c}\right)}}{(1 + \lambda^2)} \cdot \rho_{fuel}, \tag{7}$$

$$W_{wing} = 0.0051 \cdot (W_{dg} \cdot N_z)^{0.557} \cdot S_w^{0.649} \cdot AR^{0.5} + \left(\frac{t}{c}\right)_{root}^{-0.4} \cdot (1 + \lambda)^{0.1} \cdot \cos^{-1}(\Lambda_{LE}), \tag{8}$$

$$C_{Dw} = C_{D0w} + \frac{C_L^2}{\pi A R e}, \tag{9}$$

$$F_{obj_1} = C_{D0w} + \frac{C_L^2}{\pi A R e} \cdot M_{corr}, \tag{10}$$

$$F_{obj_2} = \frac{W_w}{W_{wing_initial}}. \tag{11}$$

Formally, the game is stated as shown in Eq. (4). The 2 players could play with the 5 cards ($AR, t/c, S_w, \Delta_{LE}, \lambda$) alternatively assigned to the both players, and player 1 wants to optimize the wing drag coefficient [see Eq. (10)] and player 2 the wing weight [see Eq. (11)]. The NGA optimization results are compared with the reference wing value, as shown in Fig. 2, where three solution points are marked with different shapes and colors.

The orange square and the red triangle represent two different Nash equilibrium points: the first one (game 3) represents the best point for the player 1 which minimizes wing drag coefficient with an increment in the wing weight. Vice versa the second point (game 23) minimizes the wing weight with an increment in the drag coefficient. The light blue triangle (game 30) represents one of the configurations chosen characterized by drag coefficient and wing weight values lower than the reference one. Table 2 summarizes the results of the NGA optimization, while in Figs. 3, 4 and 5 the three best wings planform compared with the reference planform (in red) are shown. The main characteristics of the reference wing are available in Table 1. The wing of the game 3 is characterized by a higher AR with respect to the wing reference value, which leads to a lower wing drag coefficient, while a lower value of the mean wing thickness percentage leads to a higher wing weight with respect to the reference one.

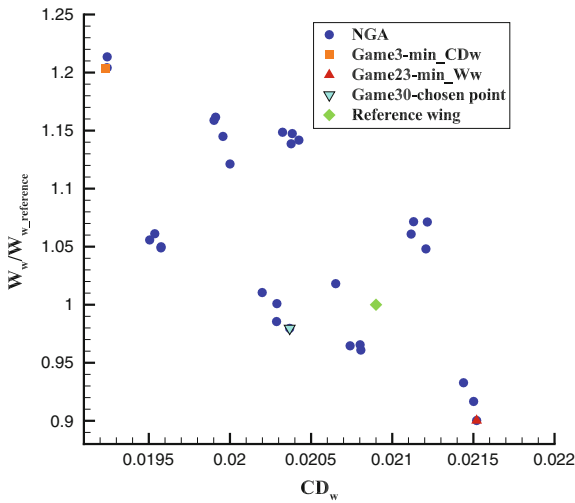


Fig. 2 NGA optimization results. The orange square (game 3) represents the best point for the player 1 which minimizes wing drag coefficient with an increment in the wing weight and the red triangle (game 23) represents the best point for the player 2 which minimizes the wing weight with an increment in the drag coefficient. The light blue triangle (game 30) represents one of the configurations chosen characterized by drag coefficient and wing weight values lower than the reference one

Fig. 3 In this figure a comparison between the wing plattform of Game 3 (blue line) and reference wing (red line) is shown. The wing configuration of Game 3 is characterized by a lower wing drag coefficient value and an higher wing weight value w.r.t. the reference wing. That is due to an higher aspect ratio value and a lower thickness ratio value than the reference wing respectively

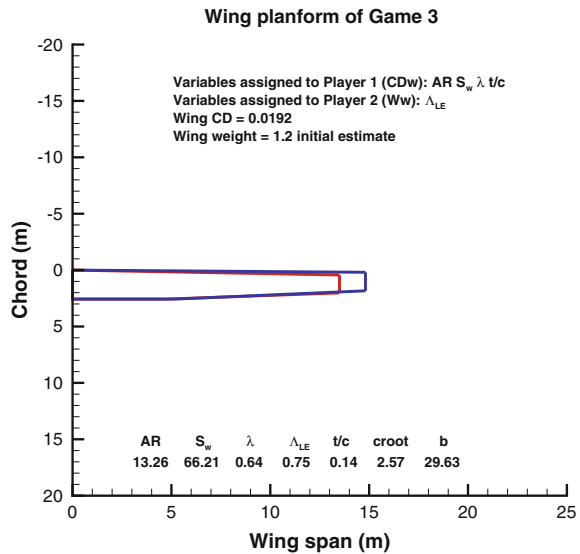
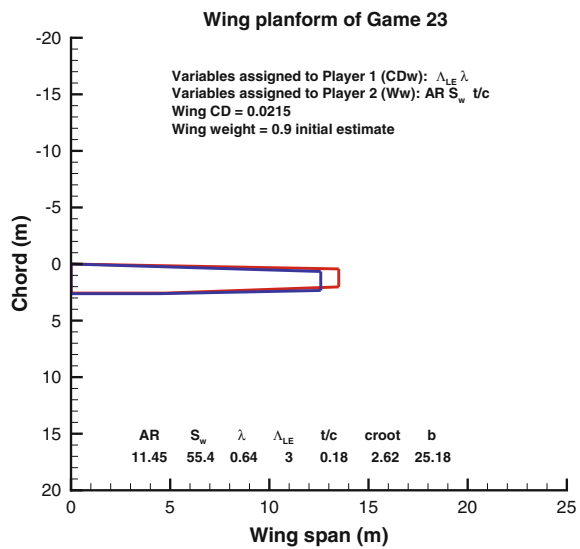


Fig. 4 In this figure a comparison between the wing plattform of Game 23 (blue line) and reference wing (red line) is shown. The wing configuration of Game 23 is characterized by an higher wing drag coefficient value and a lower wing weight value w.r.t. the reference wing. That is due to a lower aspect ratio value and an higher thickness ratio value than the reference wing respectively



The opposite reasons lead to the results obtained for the game 23. The wing of the game 30 has characteristics like the wing in game 23 but lower values of the mean thickness percentage and of the taper ratio allowing a low drag coefficient with a slightly higher wing weight.

The detailed results are reported in Table 2. The NGA optimization has been also compared with a GA scalarization and a multi-objective GA (Pareto front). In the

Fig. 5 In this figure a comparison between the wing platform of Game 30 (blue line) and reference wing (red line) is shown. The wing configuration of Game 30 is characterized by a lower wing drag coefficient value and an higher wing weight value w.r.t. the reference wing. That is due to a combination of multiple factors such as a lower value of the taper ratio and the sweep leading edge angle

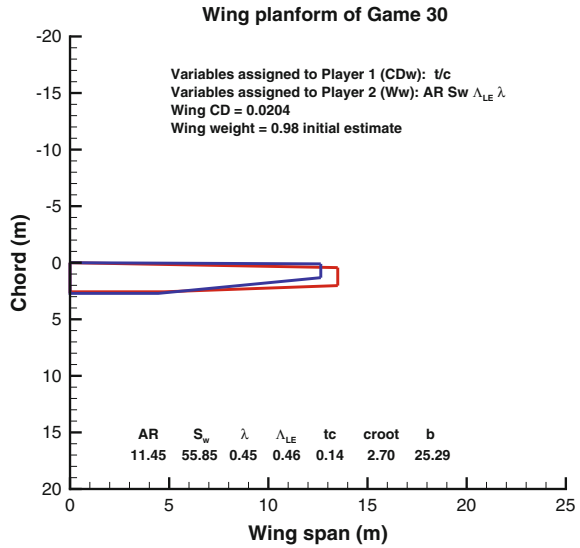


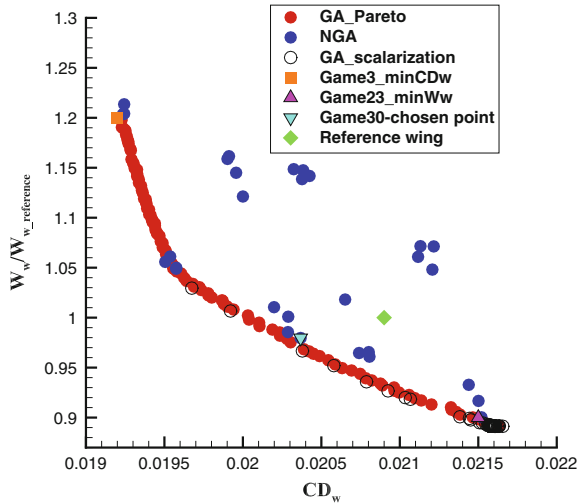
Table 1 Reference wing characteristics

	b (m)	C_{root} (m)	Λ_{LE} (deg)	λ	t/c	S_w (m^2)	MTOW (kg)
Reference wing	27	2.57	2.80	0.62	0.173	61	22215.1
C_{Dw} - wing weight @ $C_L = 0.50$	0.0209 (-) - 1048 (kg)						
Mass breakdown	W_{OE} (kg)		W_{wing} (kg)		W_{Fuel} (kg)		$W_{Payload}$ (kg)
	11917		1048		3098.1		7200

Table 2 Results of NGA application to the turboprop wing

	AR	Λ_{LE} (deg)	b (m)	λ	t/c	S_w (m^2)	MTOW (kg)
Game 3	13.26	0.75	29.63	0.64	0.14	66.21	21934.25
C_{Dw} - wing weight	0.0192 (-) - 1257.6 (kg)						
Game 23	11.45	3	25.18	0.64	0.18	55.40	21867.97
C_{Dw} - wing weight	0.0215 (-) - 943.2 (kg)						
Game 30	11.45	0.46	25.29	0.45	0.14	55.85	21465.11
C_{Dw} - wing weight	0.0204 (-) - 1027.0 (kg)						
Mass breakdown	W_{OE} (kg)		W_{wing} (kg)		W_{Fuel} (kg)		$W_{Payload}$ (kg)
Game 3	12126.60		1257.60		2607.65		7200
Game 23	11812.20		943.20		2855.77		7200
Game 30	11896.00		1027.00		2369.11		7200

Fig. 6 The results comparison among the three optimization approaches is presented. The NGA results (blue filled circle) are characterized by a good spread in the feasible zone of the Pareto front (red filled circle), while the GA scalarization points (black empty circle) are only located in the lower area of the feasible zone. The scalarization approach locates almost all the results in a bounded zone, reducing mainly the wing weight respect wing drag coefficient



scalarization optimization, GA algorithm has been chosen and the objective function is simply defined as an average weighted function, as shown in Eq. (12).

$$F_{obj} = F_{obj_1} \cdot k_{CDw} \cdot s_{CDw} + F_{obj_2} \cdot k_w \tag{12}$$

where:

- k_w , is the weight which represents the importance of the wing weight in the optimization process.
- k_{CD} , is the weight which represents the importance of the wing drag coefficient in the optimization process.
- s_{CDw} , is the scale factor useful to keep the same order of magnitude between the objective functions.

The values for the weights range from 0 to 1 for the two objective functions. In Fig. 6 the comparison between the results of the three approaches is shown, remarking a good agreement. The NGA results are characterized by a good spread in the feasible zone of the Pareto front (convexity of the Pareto front), while the GA scalarization points are only located in the lower area of the feasible zone.

Three Players – CD_w Versus W_w Versus C_{Lmax}

Starting from the two players’ optimization, a third player was involved to also consider the aircraft performance in terms of maximum wing lift coefficient. For this reason, another equation has been added [see Eq. (13)] and, consequently, the NGA was modified according Eq. (14).

$$F_{obj_3} = C_{Lmax}W \quad (13)$$

$$\begin{aligned} \Gamma &= \langle players : 3; \\ AR &: \{11.45 - 13.26\}, \\ \frac{t}{c} &: \{0.14 - 0.18\}, \\ S_w &: \{55.27 - 70.1\}, \\ \Lambda_{LE} &: \{0 - 3\}, \\ \lambda &: \{0.45 - 0.64\}; \\ C_{Dw}, W_w, C_{Lmax}W &). \end{aligned} \quad (14)$$

It must be noticed that the maximum wing lift coefficient, calculated using the Nasa-Blackwell method (Blackwell Jr. 1969), is referred to the equivalent wing. For the three players' optimization, the algorithm scans 60 possible solutions, more than the two players' optimization application, selecting only those for which the values of the objective functions are simultaneously better than the reference's wing weight and drag coefficient and greater than maximum lift coefficient. Between them the algorithm will choose the wing characterized by the maximum lift coefficient. Among all the solutions proposed by the algorithm the better compromise could be achieved assigning the leading-edge sweep angle and wing aspect ratio cards to the drag player, thickness ratio and wing area cards to the weight player, and the wing taper ratio card to the maximum wing lift coefficient player. This solution is shown in Fig. 7. The solution proposed is characterized by three players' values improved with respect to the wing reference ones. In particular, the wing drag coefficient is reduced of about 1 drag count and the wing weight of about 4%, while the maximum lift coefficient is increased of about 0.07. In Table 3 the solution proposed has been compared to the reference wing.

The application proposed in this section has been also compared with the Pareto front and the Genetic Algorithm by modifying Eq. (12) in Eq. (15)

$$F_{obj} = F_{obj_1} \cdot k_{CDw} \cdot s_{CDw} + F_{obj_2} \cdot k_w - F_{obj_3} \cdot k_{CL} \quad (15)$$

where k_{CL} is a weight which represent the importance of the wing maximum lift coefficient in the optimization process. The comparison between all the results performed by the NGA and those calculated by GA and Pareto front algorithm is shown in Figs. 8, 9 and 10. Since there are three objective functions which vary simultaneously the final results should be represented on a 3-axis graph but, to show the comparisons as well as possible, three cutting planes are presented in the figures above mentioned, focusing the attention on two pay-off functions at once. The NGA better solution (the orange square in the three figures, referred to the wing planforms shown in Fig. 7) always lies on the Pareto front, leading to comparable results among different approaches.

Fig. 7 The comparison between the wing planform of Game 25 (blue line), the three players' optimization better compromise, and the reference wing planform (red line) is shown. This compromise could be achieved assigning the leading-edge sweep angle and wing aspect ratio cards to the drag player, thickness ratio and wing area cards to the weight player, and the wing taper ratio card to the maximum wing lift coefficient player

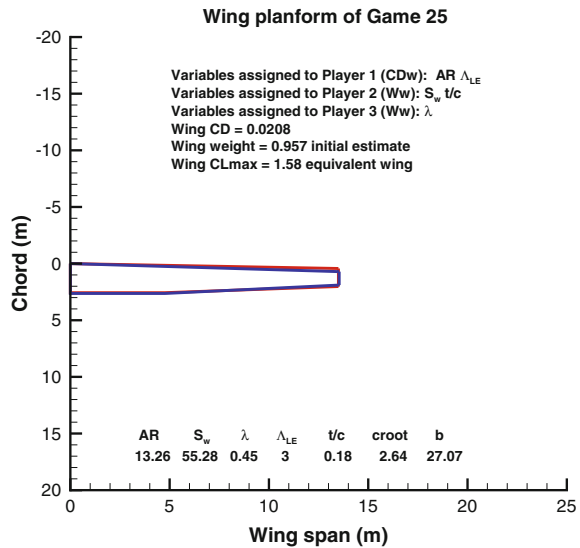


Table 3 Comparison between the reference wing and the best solution of NGA application with 3 players

	AR	Δ_{LE} (deg)	b (m)	λ	t/c	S_w (m^2)	MTOW (kg)
Reference wing	12	2.80	27	0.62	0.173	61	22215
$C_{Dw} - W_w - C_{Lmax}$	0.0209 (-) - 1048 (kg) - 1.516 (-)						
Game 25	13.26	3	27.07	0.45	0.18	55.28	21924
$C_{Dw} - W_w - C_{Lmax}$	0.0208 (-) - 1003 (kg) - 1.580 (-)						
Mass breakdown	W_{OE} (kg)		W_{wing} (kg)		W_{Fuel} (kg)		$W_{Payload}$ (kg)
Reference wing	11917		1048		3098		7200
Game 25	11872		1003		2853		7200

Conclusion

The goal of this application is to show that the Nash game theory coupled with typical genetic evolutionary algorithm, NGA, is a viable approach to use in the optimization field in order to: firstly, allow a more realistic association among variables and objective functions; secondly reduce the computational time. Moreover, the reduced distance between NGA solution points and the Pareto front attests the reasonableness and the feasibility of the results obtained. Finally, a verification of the computational time between the Pareto front, a single game of the NGA, and the GA scalarization approach has been performed on a laptop equipped with a single CPU (2.0 GHz). The elapsed time for a single NGA solution point for the 2 players application is equal to 5.14 s, for a single scalarization GA solution point is 5.91 s, and for the Pareto

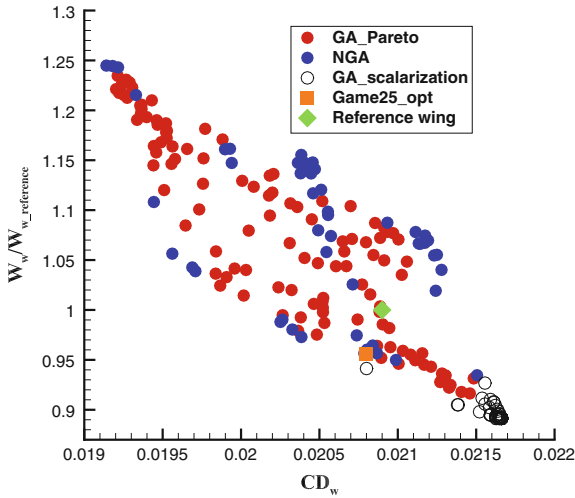


Fig. 8 The figure shows the comparison among the three optimization approaches allowing to focus the attention on two pay-off functions: W_w and CD_w . The Pareto front (red filled circle) is characterized by a more disorderly trend than the one shown in Fig. 6 because of this comparison regards just two players for an optimization process which involves three players. The comparison remarks a good agreement between the approaches. Again the scalarization approach, empty circle, locates almost all the results in a bounded zone

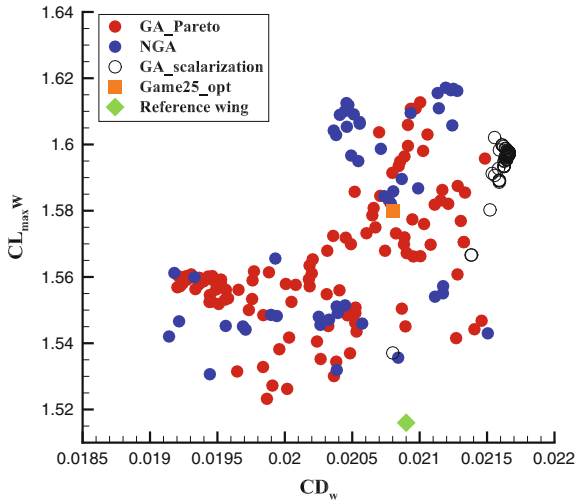
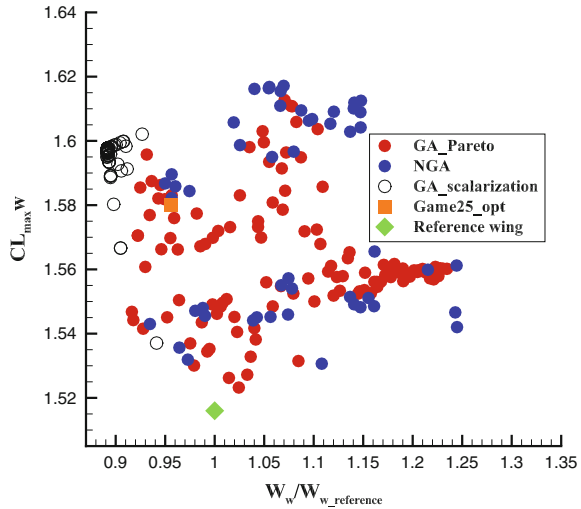


Fig. 9 The figure shows the comparison among the three optimization approaches allowing to focus the attention on two pay-off functions: $CL_{max} - CD_w$. The Pareto front (red filled circle) is characterized by disorderly trend as in Fig. 8 but by a different convexity. This difference is due to the presence of the CL_{max} objective function which must be maximize and the CD_w objective function which must be minimize. The comparison remarks a good agreement between the approaches

Fig. 10 The figure shows the comparison among the three optimization approaches allowing to focus the attention on two pay-off functions: $CL_{max} - W_w$. The Pareto front (red filled circle) is characterized by disorderly trend as in Fig. 8 but by a different convexity. This difference is due to the presence of the CL_{max} objective function which must be maximize and the W_w objective function which must be minimize. The comparison remarks a good agreement between the approaches



front is 7.57 s. The larger the number of variables or objective functions, the larger the computational time that is saved. In this case correctly design based assignment of the NGA variables to the players leads to a reduction higher than 30% in terms of computational time. As future outlook is foreseen the introduction of a higher fidelity models, like those of computational fluid dynamics (either panel based or grid resolved) and structural mechanics, to predict with a larger degree of accuracy the figures listed in Eqs. 5–11, and surrogate-based optimization strategy, to reduce evaluation function time. Moreover, the simplified models behind Eqs. 5–11 could still be applied, and extended by means of the inclusion of term-specific uncertainty factors affecting each of the figures building up the objective functions used in this optimization application. The approach is in principle easily extensible to a higher number of players, paying attention to the cards’ assignment. Further comparisons with other multi-objective optimization approaches could be investigated.

References

Basar T, Olsder GJ (1995) Dynamic non cooperative game theory. In: Classics in Applied Mathematics, 23. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA

Blackwell Jr JA (1969) A finite-step method for calculation of theoretical load distributions for arbitrary lifting-surface arrangements at subsonic speeds. Washington, D.C., Document ID: 19690021959

D’Amato E, Daniele E, Mallozzi, L., Petrone, G. (2012b) Equilibrium strategies via GA to Stackelberg games under multiple follower’s best reply. Int J Intell Syst 27, Wiley Subscription Services, Inc., A Wiley Company

D’Amato E, Daniele E, Mallozzi L, Petrone G, Tancredi S (2012a) A hierarchical multi-modal hybrid Stackelberg-Nash GA for a leader with multiple followers game. In: Sorokin A, Murphey

- R, Thai MT, Pardalos PM (eds) *Dynamics of information systems: mathematical foundations*. Springer Proceedings in Mathematics & Statistics, vol. 20, pp 267–280, Springer, New York
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, Schwefel H-P (eds) *Parallel problem solving from nature—PPSN VI*. Springer, Berlin, pp 849–858
- Della Vecchia P, Daniele E, D’Amato E (2014) An airfoil shape optimization technique coupling parsec parametrization and evolutionary algorithm. *Aerosp Sci Technol* 32(1):103–110
- Della Vecchia P, Stingo L, Corcione S, Ciliberti D, Nicolosi F, De Marco A, Nardone G (2017) Game theory and evolutionary algorithms applied to MDO in the AGILE European project. In: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. Denver, Colorado
- Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. *Evol Comput* 3:1–16
- Fudenberg D, Tirole J (1991) *Game theory*. The MIT Press, Boston
- Greiner D, Periaux J, Emperador JM, Galvan B, Winter G (2016) Nash evolutionary algorithms: testing problem size in reconstruction problems in frame structures. *ECCOMAS Congress*
- Haupt RL, Haupt SE (2004) *Practical genetic algorithms*. Wiley-Interscience, Hoboken
- Lefebvre T, Bartoli N, Dubreuil S, Panzeri M, Lombardi R, D’Ippolito R, Della Vecchia P, Nicolosi F, Ciampa PD (2017) Methodological enhancements in MDO process investigated in the AGILE European project. In: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Denver, Colorado
- Mallozzi L, Reina GP, Russo S, de Nicola C (2016) Game theoretical tools for wing design. In: Pardalos PM, Conca P, Giuffrida G, Nicosia G (eds) *Machine learning, optimization, and big data*, LNCS lecture notes in computer science, vol. 10122, pp 419–426. Springer, Berlin (ISBN 978-3-319-51468-0)
- Nash JF (1950) Equilibrium points in n-person games. *Proc Natl Acad Sci USA* 36:46–49
- Nash JF (1951) Non-cooperative games. *Ann Math Second Ser Princeton University, NJ, USA* 54:286–295
- Periaux J, Gonzalez F, Lee DSC (2015) Evolutionary optimization and game strategies for advanced multi-disciplinary design. In: *Intelligent systems, control and automation: science and engineering*. Software Pioneers. Springer, New York
- Raymer D (2002) *Aircraft design: a conceptual approach*. American Institute of Aeronautics and Astronautics Inc., Washington, D.C
- Sefrioui M, Periaux J (2000) Nash genetic algorithms: examples and applications. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. pp 509–516
- Wang JF, Periaux J, Sefrioui M (2002) Parallel evolutionary algorithms for optimization problems in aerospace engineering. *J Comput Appl Math* 149:155–169

Industrial Application of Genetic Algorithms to Cost Reduction of a Wind Turbine Equipped with a Tuned Mass Damper



Jordi Pons-Prats, Marti Coma, Jaume Betran, Xavier Roca and Gabriel Bugada

Abstract Design optimization has already become an important tool in industry. The benefits are clear, but several drawbacks are still present, being the main one the computational cost. The numerical simulation involved in the solution of each evaluation is usually costly, but time and computational resources are limited. Time is key in industry. The present communication focuses on the methodology applied to optimize the installation and design of a Tuned Mass Damper. It is a structural device installed within the tower of a wind turbine aimed to stabilize the oscillations and reduce the tensions and the fatigue loads. The paper describes the decision process to define the optimization problem, as well as the issues and solutions applied to deal with a huge computational cost.

J. Pons-Prats (✉) · M. Coma · X. Roca · G. Bugada
International Center for Numerical Methods in Engineering (CIMNE),
Universitat Politcnica de Catalunya, Campus PMT UPC,
Edifici C3, 08860 Castelldefels, Spain
e-mail: jpons@cimne.upc.edu

M. Coma
e-mail: mcoma@cimne.upc.edu

X. Roca
e-mail: xroca@cimne.upc.edu

J. Betran
Consultant on Multibody and Multiphysics Computation, Sant Cugat del Valles, Spain
e-mail: jaume.betran.prof@gmail.com

G. Bugada
Universitat Politcnica de Catalunya, Campus Nord UPC, Edificio C1,
Gran Capit s/n, 08034 Barcelona, Spain
e-mail: bugeda@cimne.upc.edu

Introduction

During the last few decades the Wind Energy industry has grown fast and engineers have designed wind turbines of increasing size, while seeking lower values of cost of energy (CoE). The sector started its industrialization in the late seventies and eighties and soon scaled the initial 50 kW, \varnothing 15 m rotors to 600 kW, \varnothing 50 m in the nineties, 3 MW, \varnothing 100 m in the 2000s and currently reaching 8 MW, \varnothing 150 m European (2012), Global (2016). Alongside the said rotor upscale, the most convenient onshore sites were taken up and modern Wind Farm developers and contractors started exploring more remote sites. Finally, during the last decade, offshore wind resources reached competitive figures of CoE in the North Sea and some smaller sites around the world. The higher and more steady offshore winds at shallow waters allowed for taking advantage of bigger rotors of an already mature technology. Offshore wind turbines present, nevertheless, particularly complex challenges in the domain of the structure dynamics additionally to the already severe wind loading. These heavy, slender structures, built on uneven seabed, have low natural frequencies that fall well within the excitation range of wave loads. In order to damp oscillations out and therefore reduce stress in the structure, some sort of absorbers are sometimes used, allowing for significant overall cost reductions. An especially interesting kind of absorber is the so called Tuned Mass Damper (TMD), composed of a massive oscillator tuned at the target frequency and a damper system to remove the energy from the resonator. The effectiveness of TMDs highly depends on its location and mass but it may have limitations due to integration issues. The present paper describes the optimization strategy and outcomes of a pre-design study of an industrial 6 MW class offshore wind turbine structure equipped with a TMD tuned at the first bending moment of the tower with a view to reduce overall structure weight and reach more competitive CoE figures. The focus is put on the strategies followed to overcome the extremely high computational time. It is directly related to the huge number of simulations accounted for the fatigue analysis for a big number of individuals dealt in a multi-objective genetic algorithms optimization scheme. The comparison of several available tools is presented. The first of them was the company's in-house optimization suite which includes gradient-based methods, plus a generic evolutionary algorithms implementation, based on NSGA-II Goldberg (1989), Deb et al. (2002), and SPEA2 Zitzler and Thiele (1998, 1999), Zitzler et al. (2001). The second tool was RMOP, CIMNE's in-house optimization platform, which implements genetic algorithm with Nash and Hybrid games Lee et al. (2009, 2010a, b, 2011a, b, c, d, 2012a, b), Periaux et al. (2009). The implementation of the GA algorithm in RMOP is quite standard. It was initially inspired on NSGAI, implementing additional functionality not only from the point of view of the evolutionary techniques, but also from the point of view of usability and user interface and the set-up of the internal parameters. To mention some of the implementations, standard techniques like SBX (Simulated Binary Cross-over) or tournament selection, were jointly added with I/O techniques and libraries to manage the definition of each individual evaluation. The key point in RMOP is the implementation of Game strategies, more specifically Nash Games, to

enhance the convergence and accuracy of the solution. This implementation leads to an hybrid definition; the Pareto optimality criteria is enriched with the information from the Nash players, so the optimization analysis benefits from the two of them Pareto and Nash.

Selection of Optimization Strategy

There are different optimization algorithms available in the company's in-house suite. In order to choose the most effective one, a comparison is conducted. This task is motivated due to the preliminary results obtained during the initial tests and discussion with company's engineers, which suggested that there are significant differences between different Multi-Objective Genetic Algorithms (MOGA) implemented within the suite, as well as the initial reservations against RMOP. The comparison is performed using mathematical test cases commonly used for this purposes. The advantages of using these cases, among others, are that are easily implemented, fast to evaluate and designed for this purpose. A preliminary TMD test case is also presented. It corresponds to a very simplified representation of the TMD, but the main aim when analyzing this particular test case is to anticipate potential issues both from the implementation viewpoint and from the results viewpoint. The studied MOGAs are the 3 available in the suite, plus RMOP:

- Evolution: it is a generic implementation of an evolutionary algorithm. It is quite a simple implementation with a limited control over the setup parameters of the algorithm.
- NSGA2: it is a well known algorithms developed by Prof. Deb et al. (2001b). Its applicability and high performance have been documented widely.
- SPEA2: it is a well known evolutionary algorithm developed by Zitzler and Thiele (1998).
- RMOP: Genetic algorithms with game theory. It is an in-house CIMNE development which combines some basic evolutionary algorithms strategies with Nash games for improved convergence and accuracy. It is possible thanks to the combination of Pareto optimality criteria with Nash Games as previously described.

The selected test cases are:

- KUR; a mathematical test case with 3 design variables and 2 objective functions. Its complexity comes from the definition of the functions.
- TNK mathematical test case with 2 design variables, 2 objective functions and 2 constraints, it is a first step into a restricted search space.
- CPT3 mathematical test case with 2 design variables, 2 objective functions and 1 constraint, its complexity is a combination between the objective functions and the restricted search space.
- OSY mathematical test case with 6 design variables, 2 objective functions and 6 constraints, which defines a very restricted search space.

- ZDT2; a mathematical test case with 30 design variables and 2 objective functions. Part of its complexity comes from the number of design variables.
- LZ09-F1 mathematical test case with 1 objective function and a variable number of design variables. This characteristics makes it interesting for constantly increase the problem complexity.
- TMD test case, a multi-objective and multi-disciplinary structural problem based on the real-world case of designing a wind turbine.

For more details about the mathematical test cases, please refer to Deb et al. (2001a, b, 2002), Chafekar et al. (2003), Deb and Goel (2002).

Comparison of the Mathematical Test Cases Results

For a fair comparison, a common set-up were defined for all the algorithms and all the mathematical test cases. This common set-up defines a population size equal to 4 times the number of design variables, and a number of maximum evaluations equal to hundred times the population size. The crossover probability was defined equal to 0.9 and the mutation probability equal to 0.1. Figures 1, 2, 3, 4, 5 and 6 show a comparison between the results obtained by each algorithm. Figures 1, 3 and 5 show the convergence history of the objective functions for each test case. In all the three cases, RMOP and NSGA2 are the ones converging the faster and lower. On the other hand, Figs. 2, 4 and 6 show the Pareto fronts for each of the three cases. It is clear that the number of evaluations is not enough to fully capture the front shapes with enough accuracy, but, due to the fact that the aim of the analysis was to detect which algorithms performs the better with a restricted number of evaluations, then the objective was fully fulfilled.

In an overall performance analysis of the results, RMOP presents a better average performance. Evolution algorithm shows poor performance in all tests done, both in the convergence of the fitness functions and in the capture of the Pareto Front. NSGA2 shows results compatible with RMOP results in most of the problems and in most of the Pareto front regions. However, RMOP better captures the Pareto Front in all the cases. SPEA2 shows results compatible with RMOP results in some of the problems and in most of the Pareto front regions. The general performance is lower than RMOP. In some test cases (TNK and OSY mainly), Pareto front regions are not well populated when using NSGA2 and SPEA2. It is clear that this phenomena is a direct consequence of the imposed limitation on the number of evaluations and the size of the population. It was an expected drawback which was accepted for the sake of saving time. In case the company's proprietary software would be a requirement, NSGA2 method should be the most appropriate selection. In case company's software can be coupled with the RMOP optimization algorithm, then this configuration is the best choice.

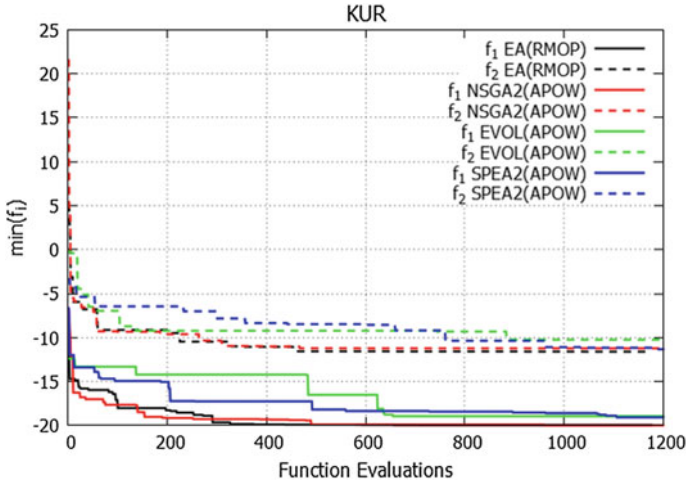


Fig. 1 KUR test case convergence

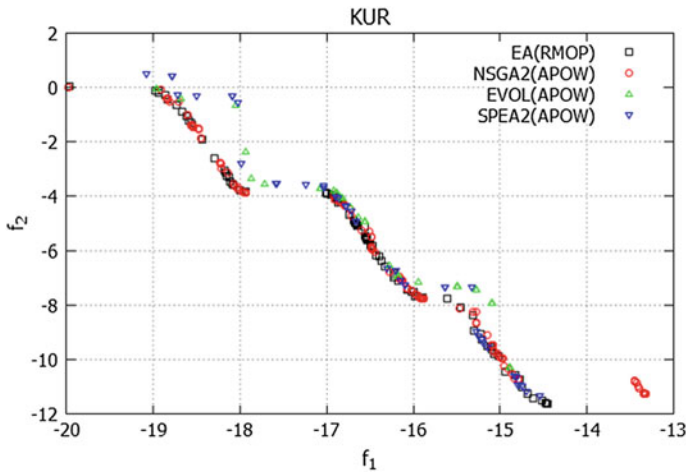


Fig. 2 KUR Pareto front

Industrial Application: TMD Optimization

The industrial application is based on a real case, a set of wind turbine, nacelle, tower and mono-pile which the company is designing and manufacturing for an Atlantic offshore site. The addition of a TMD is under study. The wind turbine design including rotor, nacelle and tower is fixed, so the analysis will not modify

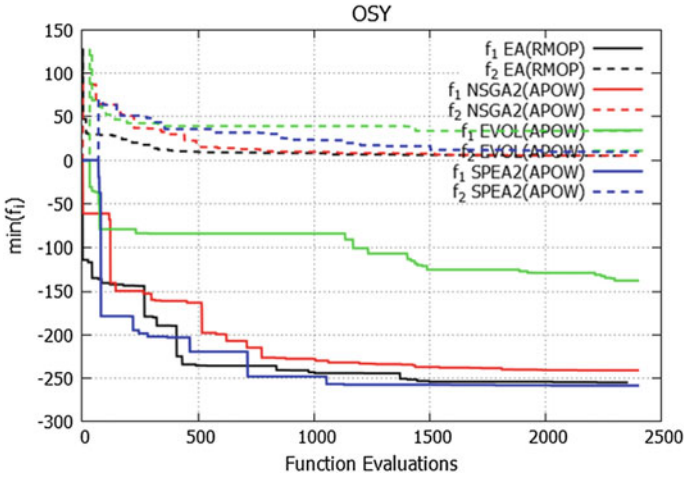


Fig. 3 OSY test case convergence

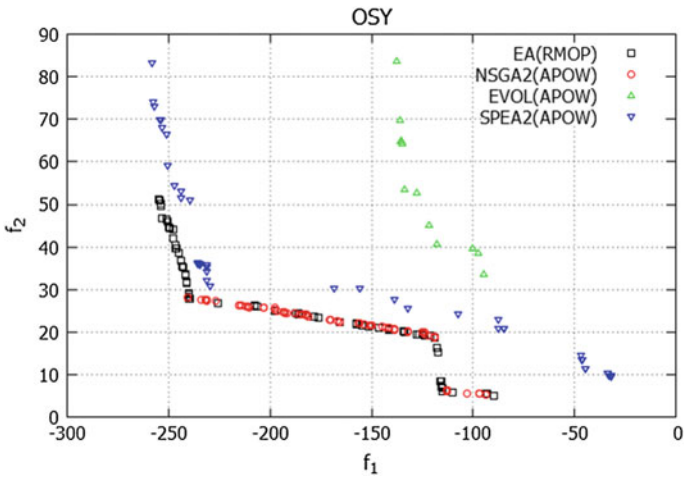


Fig. 4 OSY test case Pareto front

any of their parameters. The main aims of the work is to optimize the mono-pile and the TMD. Two are the principal objective functions; the first of them is the structural performance of the structure when using or not the TMD, and according to the mass and dumping parameters, while the second one is the overall cost including the mono-pile and the TMD. The performance is split into three objective functions representing the behavior under ultimate and fatigue loads. To define the performance function,

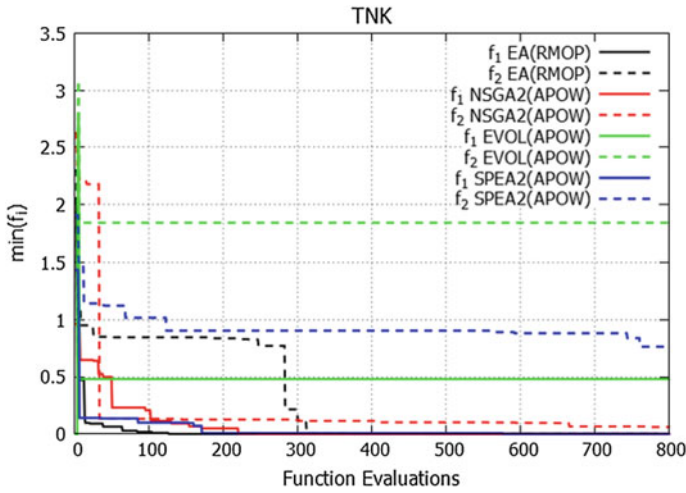


Fig. 5 TNK test case convergence

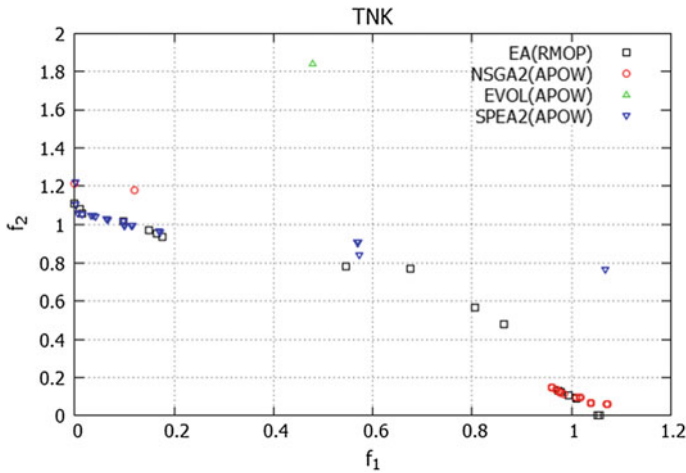


Fig. 6 TNK test case Pareto front

the objective functions that reflect performance in Ultimate and Fatigue loads must be defined:

$$\begin{aligned}
 FO_{FLS-fa} &= w_1^F \cdot M_{x,1}^F + w_2^F \cdot M_{x,2}^F + w_3^F \cdot M_{x,3}^F + \\
 &\quad + w_4^F \cdot M_{x,4}^F \\
 FO_{FLS-ss} &= w_1^F \cdot M_{y,1}^F + w_2^F \cdot M_{y,2}^F + w_3^F \cdot M_{y,3}^F + \\
 &\quad + w_4^F \cdot M_{y,4}^F \\
 FO_{ULS} &= w_1^U \cdot M_{xy,1}^U + w_2^U \cdot M_{xy,2}^U + w_3^U \cdot M_{xy,3}^U + \\
 &\quad + w_4^U \cdot M_{xy,4}^U
 \end{aligned}
 \tag{1}$$

Being

- FO_{FLS-fa} : fore-aft FLS performance objective function,
- FO_{FLS-ss} : side-to-side FLS performance objective function,
- FO_{ULS} : fore-aft FLS performance objective function,
- $M_{x,i}$: moment in fore-aft direction at point i,
- $M_{y,i}$: moment in side-to-side direction at point i,
- $M_{xy,i}$: modulus of resultant moment

where i represents the points: 1 for the Tower bottom, 2 for the Tower lower intermediate, 3 for the Tower upper intermediate, and 4 for the Tower top. Variables w_i^U and w_i^F are weights to take into account the more relevance of the moments when closer to the bottom of the tower:

$$\begin{aligned} w_i^U &= \frac{1}{1+a \cdot i} \\ w_i^F &= \frac{1}{1+a' \cdot i} \end{aligned} \tag{2}$$

Parameters a and a' are chosen according to structural criteria. For this case, in order to obtain a linear cost function and under agreement with the engineers of the company, they are chosen constants and equal to 0.1. Finally, objective functions in Ultimate and Fatigue results in:

$$\begin{aligned} FO_{FLS-fa} &= \frac{M_{x,1}^F}{1.1} + \frac{M_{x,2}^F}{1.2} + \frac{M_{x,3}^F}{1.3} + \frac{M_{x,4}^F}{1.4} \\ FO_{FLS-ss} &= \frac{M_{y,1}^F}{1.1} + \frac{M_{y,2}^F}{1.2} + \frac{M_{y,3}^F}{1.3} + \frac{M_{y,4}^F}{1.4} \\ FO_{ULS} &= \frac{M_{xy,1}^U}{1.1} + \frac{M_{xy,2}^U}{1.2} + \frac{M_{xy,3}^U}{1.3} + \frac{M_{xy,4}^U}{1.4} \end{aligned} \tag{3}$$

The functions will be minimized in order to maximize the performance of the system.

The cost is the main objective function, because the benefits of the company is directly related to the manufacturing and installation cost (CAPEX cost) of the TMD. It can be calculated according a complete cost function, or just considering the cost of the TMD (its mass as the main contributor to the cost).

$$\begin{aligned} FO_{COST} &= 74600 + 1.415 \cdot TMDmass + \\ &+ 11700 + 29000ifTMDmaxEx < 0.5 \\ &74600 + 1.415 \cdot TMDmass + \\ &+ 44000 \cdot TMDmaxEx + 11700 + \\ &+ 29000ifTMDmaxEx > 0.5 \end{aligned} \tag{4}$$

Due to the cost of computing a single individual, several strategies have been implemented to reduce the overall computational cost. These strategies include on one hand stopping the calculation if the individual is not fulfilling the restrictions, and on the other hand a careful selection of the load cases to be calculated, just to mention two of them. Although applying these simplifications, the evaluation workflow is quite complex involving several solvers and checkpoints.

Three different models based on three different loads' computation approaches have been studied. Based on accuracy reasons and on the possibility of customization of the tool, a FE based flexible Multibody model of a wind turbine, substructure and tuned mass damper built in SAMCEF was considered in the first place. This option was early discarded for the full optimization procedure due to the high CPU times, which were unfordable when considering the industrial cost limitation. This limitation is related, amongst others, to the limited number of software licenses available, which is not a technical issue, but an industrial issue. From the technical point of view, the license issue was limiting the parallelization of the individual evaluations. Its use was reduced to periodic verification purposes only. GH Bladed was chosen as an alternative. The CPU time per thread is similar to that of SAMCEF but the available licenses at the company allowed for multiple simulations running in parallel in different threads, which significantly speed up the overall optimization procedure. While SAMCEF is a FE solver that features mechanism modeling Siemens LMS Samtech (2012), Bladed is a Wind Turbine dedicated multi-body simulation (MBS) software that models mechanisms with kinematic laws and features elasticity by modal condensation of its main structures Bossanyi (2010). This approach has an impact on accuracy of the results, mostly due to poor modeling of TMD nonlinear region, that is overcome with periodic verification with a higher standard approach. A third option is finally considered which stretches the latter approach. An ad-hoc solver is developed to compute loads of a simplified model of WT, substructure and nonlinear TMD. While the motion of the TMD remains in the linear region a fast and exact recursive closed form solution is used Betran and Breuker (2014) and it swaps to Newton family solvers, HHT, when nonlinearities must be accounted for. This approach totally solves the problem of threads used in parallel and the CPU cost per load case is significantly reduced. The use of the SAMCEF model for verification guarantees accuracy of overall procedure. Finally, the selected procedure was a mix between the use of Bladed and the ad-hoc solver, which was implemented within MATLAB. Bladed was used to perform a Campbel analysis of the individual, which first determine its feasibility, and early discard those leading to a poor design.

Figure 7 describes the individual evaluation workflow. Step by step can be described as:

1. START: Start node for the individual evaluation.



Fig. 7 Evaluation workflow

2. `getTempDir`: Scripting Process (Java). Sets the local variable `TempDir` with the path of the current evaluation folder. It changes for each individual evaluation of the optimization.
3. `Init`: Scripting Process (Python). Sets initial values for several variables. The relation of initiated variables.
4. `Copy Essential Files`: File Copy Process. Copy the files needed for the evaluation to the temporary folder, which must be located in the optimization directory: (a) Excel file (`DLCs.xlsx`) containing the dynamic load cases to be evaluated. (b) Bladed Campbell model file (`DTBLADED.MODEL`). (c) Binary Matlab file containing information about the files and folders name (`Names.mat`).
5. `Eval.DVs`: File Creation Process. Creates a file containing the value of each design variable. The file format is ASCII with one line containing: `TMDmass`, `TMDfreq`, `TMDdamp`, `TMDnode`.
6. `subsBladedFile`: Execution Process. Runs executable file `subsBladedInFile.exe`. It combines the files `DTBLADED.MODEL` with `Eval.DVs` to generate `DTBLADED.IN`.
7. `Dir BladedRun`: Directory Creation Process. Creates the directory `BladedRun` inside the `TempDir` directory so `Bladed` can run in it.
8. `Dtbladed.exe`: Execution Process. Runs executable file `dtbladed.exe` to perform the Campbell analysis. The call does not take parameters, it is: `BladedFolder dtbladed.exe`.
9. `evalBladedCampbel`: Scripting Process (Python). Reads Campbell results from file `BladedRun/modalResultsFileName` and evaluates the 3P dynamic criteria. It also sets the Penalty output variable to 1 if the file does not exist or to 2 if the dynamic criteria is not met.
10. `Cond evalBladedCampbel`: Condition. Decision point that checks if the evaluation of the Campbell analysis is satisfactory (`ValEvalBladedCampbel == 0`) and then runs the node `evalMatlab.exe` or it skips further evaluations and goes directly to the node `Delete TempDir`.
11. `evalMatlab.exe`: Execution Process. Runs `evalMatlab.exe`. It calculates the dynamic load cases transformation for the current TMD, the performance objective functions, among others. Call is: `ToolsFolder/evalMatlab.exe TMDmaxExcursionRestr ApplyExcursionBreak`.
12. `Read Constraint`: Parameter Reader. Reads `Eval.constraint` file previously written by `evalMatlab.exe` which contains the value of `TMDExcursionRestr`.
13. `Calc TMDmaxExcursion`: Calculator Process. Calculates the value of the maximum excursion: $TMDmaxExcursion = TMDmaxExcursionRestr \cdot TMDExcursionRestr$.
14. `Eval TMDmaxExcursionRestr`: Condition. Decision point for maximum excursion criteria.
15. `Read Performance FOs`: Parameter Reader. Reads `Eval.2to4individual` file which contains the values of the objective functions from 2 to 4. The file format is ASCII with one line containing: $FO(FLS - fa)$, $FO(FLS - ss)$, FO_{ULS} .
16. `TMD cost FO calc`: Scripting Process (Python). Calculates TMD cost objective function. See [1] for details on the function.

17. ErasePenalty: Calculator Process. Sets local variable Penalty value to 0, indicating that the evaluation is correct and have not been applied any penalty during the process.
18. Delete TempDir: Scripting Process (Python). If the evaluation has been launched from an optimization process deletes TempDir folder, if it has been launched for a single evaluation it does not erase the TempDir.
19. FINISH: Finish node for the individual evaluation. MarcaPenalty3: Calculator Process. Sets local variable Penalty value to 3, indicating that the evaluation has been stopped during the maximum excursion check.
20. MarcaPenalty3: Calculator Process. Sets local variable Penalty value to 3, indicating that the evaluation has been stopped during the maximum excursion check.

An initial definition of the optimization problem had 8 design variables, 4 to define the basic substructure geometry and 4 to define the TMD characteristics. The variation of substructure geometry has a direct impact on the driving substructure cost and on the dynamic behavior of the whole system. The variation of the TMD characteristics contribute for each individual to reduce the dynamic response and has a secondary contribution to the cost. Finally, a selection of 4 design variables was made to simplify the problem, all of them related to TMD. This selection includes the mass, the frequency, the damping coefficient and the station where to install it. Objective functions has been described in (3) and (4). Both the search space and the solution space are multi-dimensional, which means the Pareto front is no longer a 2D line, nor a 3D surface. The analysis of the solution is done on the projection of this multi-dimensional space into 2D plots. Figures 12, 13 and 14 show an example of how the solutions are plotted, taking couples of the 4 objective functions. Additionally, Figs. 15 and 16 show the Pareto Front plotting 3 of the objective functions, one of them as a color scale. Figures 8, 9, 10, and 11 are the convergence history of each objective function. No major issues can be extracted from these plots, more than compare how fast each function is converging. Figure 8

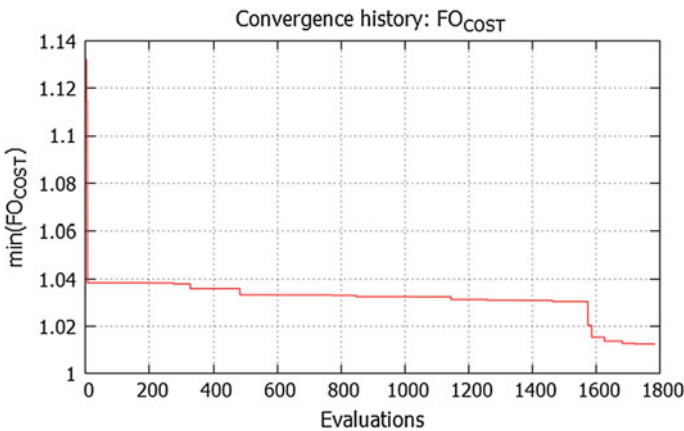


Fig. 8 Cost objective function convergence

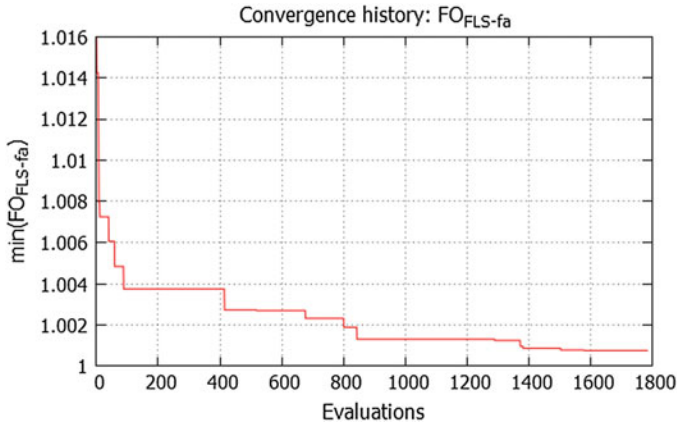


Fig. 9 Fatigue loads (front-aft) objective function convergence

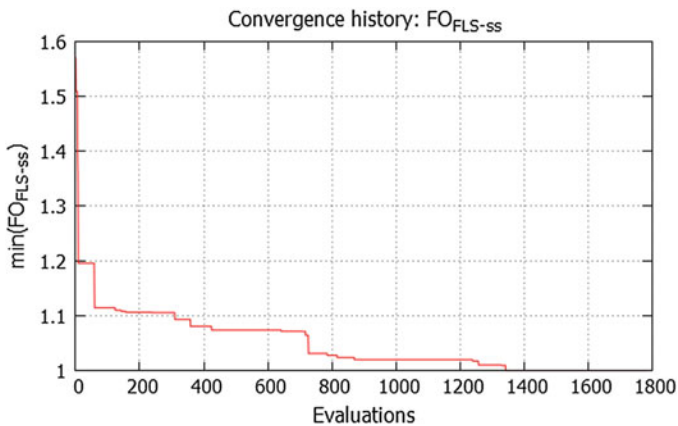


Fig. 10 Fatigue loads (side to side) convergence

shows the cost objective function convergence, which until the end of the analysis does not present a significant improvement. Figure 9 shows a gradual improvement of the Fatigue loads (front aft). Figure 10 shows also a constant improvement of the function, anyway, the last 200 evaluations do not provide further improvement. It shows two regions, the first half with a gradual improvement, and the second half where the improvement is less significant. Figure 11, regarding the ultimate loads, does not show improvement much improvement during the optimization. Although the scale of the y axis has been normalized, to fulfill with the NDA signed with the company, all the functions show a good improvement along the optimization analysis, compared to initial values. Later, when analyzing the Pareto Front plots, a comparison with the baseline design will be provided.

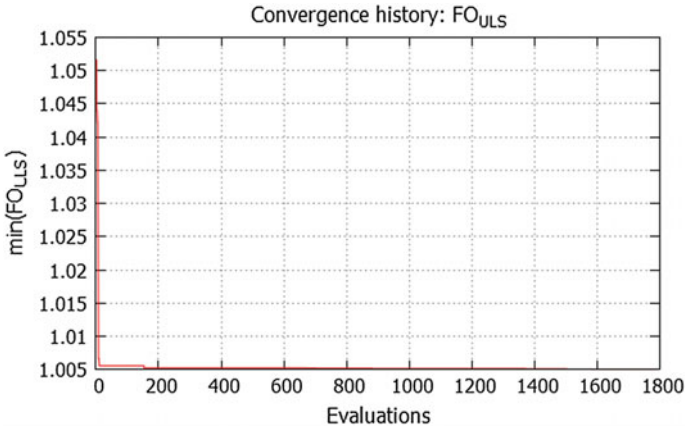


Fig. 11 Ultimate loads objective function convergence

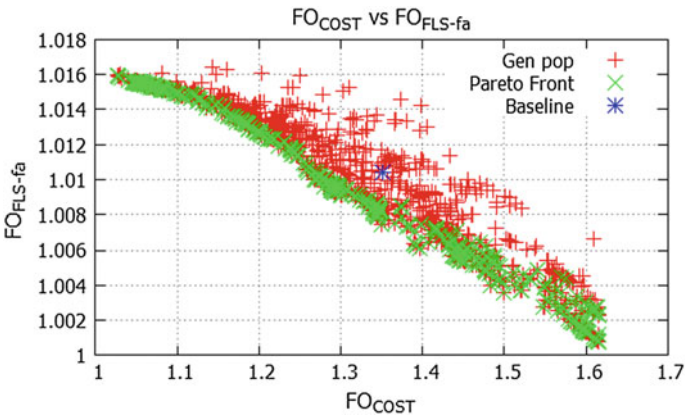


Fig. 12 Cost versus fatigue loads (front-aft)

The plots for the Pareto Front show the results using Bladed plus MATLAB implementation. In all of them an improvement respect to the baseline design is obtained. There are a lot of Pareto individuals improving the values for the baseline, demonstrating the performance of the optimizer RMOP as expected. The computational cost associated to this analysis is reduced applying a preliminary selection of the load cases for each individual, without penalizing the accuracy and feasibility of the design. Figure 12 shows the cost versus the Fatigue loads. As shown in the graph both functions are opposed; to improve one function the other must get worse. This is not always the case, as happens with the two functions related to fatigue loads. Individuals that belong to the Pareto Front are marked in green. As a remark, the plots presented are a projection of the four dimensional objective functions space; this yields to a Pareto Front representation that has individuals that may appear as

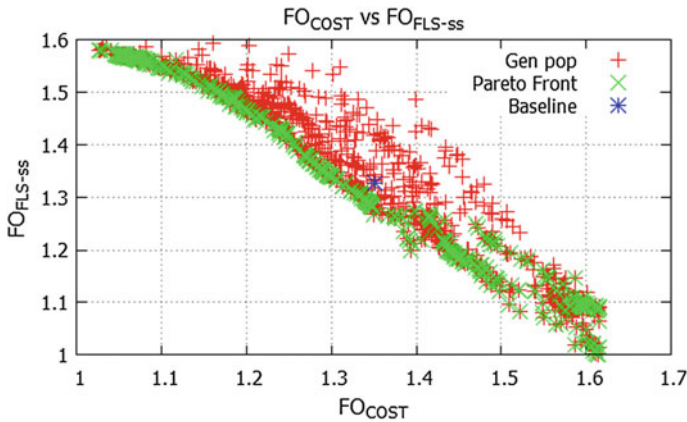


Fig. 13 Cost versus fatigue loads (side to side)

dominated, but are not. Those individuals appear as non-dominated in other projections. TMD cost in front of PF Fatigue side-side is shown in Fig. 13. These two functions are opposed too. TMD cost in front of PF Ultimate is shown in Figure 14. The two functions are opposed, but the more expensive TMDs is, the PF Ultimate almost constant is. the Pareto Front of the two Fatigue loads (front aft and side to side) is not shown because the two functions present a strong correlation, so any improvement on one of the two also means improving the other one. Two additional Pareto Front plots, including 3 objective functions, are presented. Figures 15 and 16 do not shown the dominated points to simplify and make them more understandable. In the second one, the region where the ultimate loads keep constant while the cost or the fatigue loads increase is also there, as seen previously in Fig. 14. In the Pareto front plots, some individuals have been marked as “selected”. Those individuals have been used by the company to evaluate the overall performance of the optimization and to compare the the four objective functions values with the baseline design. All the selected points are located near the influence area of the baseline, although they improve the baseline values. It is true that in some cases, it is not possible to simultaneously improve all the functions, but improvements of about 10 to 20% are possible. Table 1, describes the error of the selected individuals compared to the baseline. The individuals improving all the objective functions show lower improvement, while those with larger improvements in some functions show more variability on the error along the four objective functions.

It has been mentioned that several strategies have been evaluated in order to reduce the computational cost associated to the analysis. The most relevant ones have been:

- Parallelize: it is the first one thing about, but one should bear in mind that the solver can be also parallelized. The use of smart strategy, defining how many cores is using each individual evaluation and how many are available, is of great

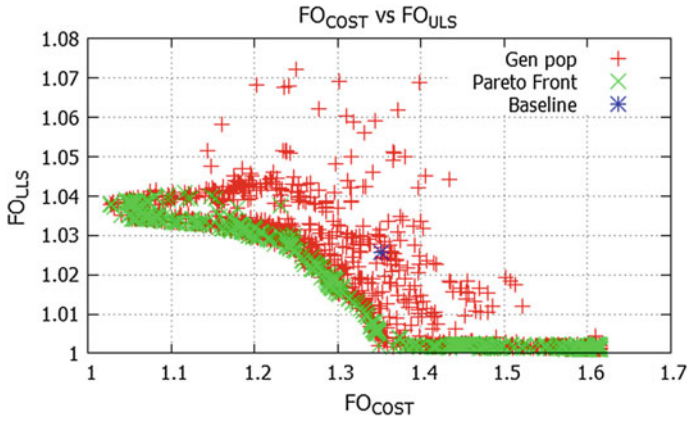


Fig. 14 Cost versus ultimate loads

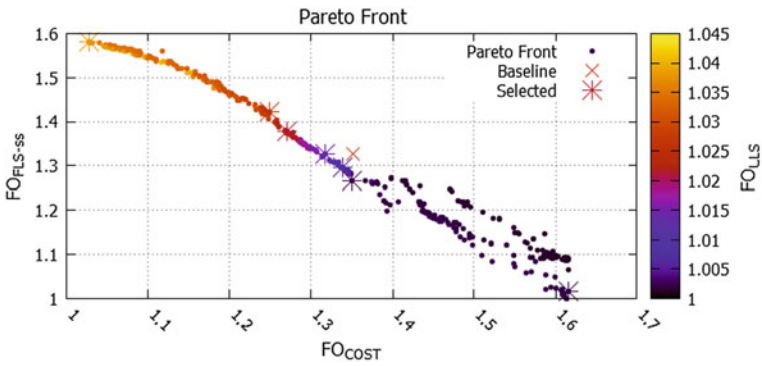


Fig. 15 Cost versus fatigue loads (side to side) vs ULS

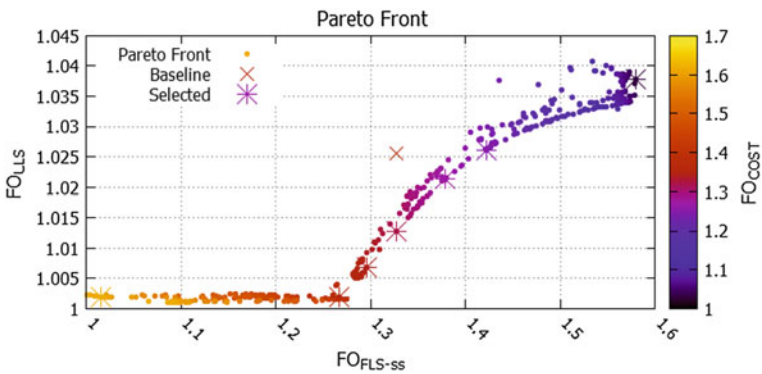


Fig. 16 Fatigue loads (side to side) versus ultimate loads versus cost

Table 1 Relative error compared to baseline

Cost	FLS-fa	FLS-ss	ULS
-24.00	-0.55	19.03	1.18
-0.13	-0.29	-4.58	-2.31
-2.52	-0.14	-0.004	-1.25
19.54	-0.80	-23.49	-2.31
-7.56	0.09	7.15	0.04
-0.90	-0.23	-2.35	-1.84
-6.04	-0.04	3.83	-0.42

importance. This is the first and more important strategy because it can be applied whatever analysis and solver you are going to use.

- **Individual evaluation:** in some cases it is not possible to interact with the individual evaluation, so its cost cannot be reduced. But it was not the case of the actual analysis. The standard procedure of the company, when validating a design includes a long list of load cases to be evaluated. Initially, the company was requesting to apply the same list to each individual on the evaluation, leading to an unaffordable cost. A careful analysis determined that the relevant load cases can be restricted to only a 5%, then the computational cost has been extremely reduced.
- **Constraints:** one can use the constraints as restrictions, so defining a go-no go criteria. If any individual does not fulfill a constraint, the individual is penalized and the evaluation is stopped. This strategy is easy to implement if your evaluation workflow is split into several steps, otherwise it can be difficult or no sense to apply because the cost reduction is not relevant.

An important point must be highlighted in regards the restrictions applied to the individuals. There have been 137 individuals that do not meet the maximum excursion for the TMD displacement (set at 1.5 m), with a maximum excursion of 2.1 m. Another remark is that any individual has been penalized for the dynamic check, evaluated with the Campbell analysis and the 3P criteria. This should be analyzed further to evaluate the reason that any individual has not met the constraint; maybe is not set correctly or maybe there is some error in the implementation. If it is confirmed that the restriction was satisfactorily setup, then the use of Bladed, to perform the Campbell analysis should be removed, simplifying the workflow and reducing the computational cost of each individual evaluation.

Conclusions and Further Work

Industrial applications differ from academic problems on many ways. Although academic and mathematical test case can reach a high complexity level, the multidisciplinary involved in industrial problems, added to the complexity of the design

process by itself conform the key issues. This communication is aimed to describe how the authors deal with this complexity, and how closely working with the engineers in the company face and partially solve this problem. The paper is also aimed to demonstrate the capabilities of RMOP in comparison to industrial suites. The results from both the mathematical test cases and the industrial application show how RMOP performs better, even without using additional functionalities like Nash Games. From the point of view of the industrial application, the results lead to a relevant reduction of the cost, as well as the loads. The Pareto hyper-surface is defining a large number of Pareto individuals which are improving the 4 objective functions in comparison with the baseline design. Significant improvements have been obtained when improving. Although those individuals with spectacular improvements on one function do not show consistent improvements for all the four objective functions, there are many individuals improving the four functions within the range of 1–5%, which is more than interesting. From the point of view of the company, the cost of the TMD was the most important objective function, so it was used to identify and select those individuals and more promising configurations. Further work is two-fold. On one hand an on-going implementation of a most simplified solver in MATLAB, which can lead to a simplified solver, with a reduced computational cost but with an appropriate accuracy level. On the other hand, CIMNE is working on the continuous improvement of RMOP. It focus on the implementation of hierarchical evaluation strategies within the platform. Each of the two will mean a significant improvement on the calculation time and on the efficient use of the available solver to get a fast scan of the search space and an accurate optimal solution.

Acknowledgements The authors would like to thank the company to provide the opportunity to publish the results of the join work, although due to the non-disclosure agreement between to two parts, the name of the company and the details and values have been masked. The authors also acknowledge the engineers of the company in structural and aerodynamic loads departments for their support during the implementation of the analysis.

References

- Betran J, De Breuker R (2014) Exact method for coupled non-linear state-space simulation and its application to a flexible multibody wind turbine aeroelastic code. *J Aeroelasticity Struct Dyn* 3(2)
- Bossanyi EA (2010) Bladed theory manual version 4
- Chafekar D, Xuan J, Rasheed K (2003) Constrained multi-objective optimization using steady state genetic algorithms. Springer, Berlin, Heidelberg, pp 813–824
- Deb K, Goel T (2002) Multi-objective evolutionary algorithms for engineering shape design. Springer, US, Boston, MA, pp 147–175
- Deb K, Pratap A, Meyarivan T (2001a) Constrained test problems for multi-objective evolutionary optimization. In: International conference on evolutionary multi-criterion optimization. Springer, pp 284–298
- Deb K, Pratap A, Meyarivan T (2001b) Constrained test problems for multi-objective evolutionary optimization. Springer, Berlin, Heidelberg, pp 284–298
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evol Comput IEEE Trans* 6(2):182–197

- European Wind Energy Association et al (2012) Wind energy-the facts: a guide to the technology, economics and future of wind power. Routledge, Abington
- Global Wind Energy Council GWEC (2016) Global wind statistics 2015
- Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, Massachusetts
- Jacques P, Lee DS, Gonzalez LF, Srinivas K (2009) Fast reconstruction of aerodynamic shapes using evolutionary algorithms and virtual nash strategies in a cfd design environment. *J Comput Appl Math* 232(1):61–71
- Lee D-S, Gonzalez LF, Periaux J, Srinivas K (2009) Evolutionary optimisation methods with uncertainty for modern multidisciplinary design in aeronautical engineering. In: 100 volumes of notes on numerical fluid mechanics. Springer, pp 271–284
- Lee DS, Periaux J, Pons-Prats J, Bugada G, Oñate E (2010a) Double shock control bump design optimization using hybridised evolutionary algorithms. In: Evolutionary computation (CEC), 2010 IEEE congress on. IEEE, pp 1–8
- Lee DS, Srinivas K, Gonzalez LF, Periaux J, Obayashi S (2010b) Robust multidisciplinary design optimisation using cfd and advanced evolutionary algorithms. *Computational fluid dynamics review 2010*. World Scientific Publishing Company, Incorporated, pp 469–491
- Lee DS, Gonzalez LF, Periaux J, Srinivas K (2011a) Efficient hybrid-game strategies coupled to evolutionary algorithms for robust multidisciplinary design optimization in aerospace engineering. *IEEE Trans Evol Comput* 15(2):133–150
- Lee DS, Gonzalez LF, Periaux J, Srinivas K, Onate E (2011b) Hybrid-game strategies for multi-objective design optimization in engineering. *Comput Fluids* 47(1):189–204
- Lee D-S, Periaux J, Onate E, Gonzalez LF, Qin N (2011c) Active transonic aerofoil design optimization using robust multiobjective evolutionary algorithms. *J Aircr* 48(3):1084–1094
- Lee DS, Gonzalez LF, Periaux J, Bugada G (2011d) Double-shock control bump design optimization using hybridized evolutionary algorithms. *Proc Inst Mech Eng Part G J Aerosp Eng* 225(10):1175–1192
- Lee DS, Periaux J, Gonzalez LF, Srinivas K (2012a) Robust multidisciplinary uas design optimisation. *Struct Multi Optim* 45(3):433–450
- Lee DS, Morillo C, Bugada G, Oller S, Onate E (2012b) Multilayered composite structure design optimisation using distributed/parallel multi-objective evolutionary algorithms. *Compos struct* 94(3):1087–1096
- Siemens LMS Samtech (2012) In: Siemens LMS samtech (ed.) Samcef user manual 14:1. LMS Samtech, Belgium
- Zitzler E, Thiele L (1998) An evolutionary algorithm for multiobjective optimization: the strength pareto approach
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
- Zitzler E, Laumanns M, Thiele L et al (2001) Spea2: Improving the strength pareto evolutionary algorithm

Part IV
Optimization Under Uncertainty

Aerodynamic Shape Optimization by Considering Geometrical Imperfections Using Polynomial Chaos Expansion and Evolutionary Algorithms



Athanasios G. Liatsikouras, Varvara G. Asouti,
Kyriakos Giannakoglou and Guillaume Pierrot

Abstract Uncertainties, in the form of either non-predictable shape imperfections (manufacturing uncertainties) or flow conditions which are not fixed (environmental uncertainties) are involved in all aerodynamic shape optimization problems. In this paper, a workflow for performing aerodynamic shape optimization under uncertainties, by taking manufacturing uncertainties into account is proposed. The uncertainty quantification (UQ) for the objective function is carried out based on the non-intrusive Polynomial Chaos Expansion (niPCE) method which relies upon the CFD software as a black-box tool. PCE is combined with an evolutionary algorithm optimization platform. CAD-free techniques are used to control the shape and simultaneously generate shape imperfections; next to this, a morphing/smoothing tool adapts the CFD mesh to any new shape. In the cases presented in this paper, all CFD evaluations are performed in the OpenFOAM environment.

Introduction

A variety of stochastic and gradient-based optimization methods have been developed to cope with shape optimization problems in aerodynamics. Most of the relevant algorithms minimize (or maximize) an objective function (to be denoted as F)

A. G. Liatsikouras (✉)

ESI Software Germany GmbH, Kruppstr. 90, 45145 Essen, Germany

e-mail: athanasios.liatsikouras@esi-group.com; thanasis.liatsi@gmail.com

V. G. Asouti · K. Giannakoglou

National Technical University of Athens (NTUA), School of Mechanical Engineering,

Parallel CFD and Optimization Unit, Athens, Greece

e-mail: vasouti@mail.ntua.gr

K. Giannakoglou

e-mail: kgianna@central.ntua.gr

G. Pierrot

ESI-Group, 99 Rue des Solets, 94513 Rungis, France

e-mail: guillaume.pierrot@esi-group.com

© Springer International Publishing AG, part of Springer Nature 2019

E. Andrés-Pérez et al. (eds.), *Evolutionary and Deterministic Methods for Design Optimization*

and Control With Applications to Industrial and Societal Problems, Computational

Methods in Applied Sciences 49, https://doi.org/10.1007/978-3-319-89890-2_28

assuming that the flow conditions are fixed and/or the exact geometry can be manufactured. However, this is not the case in real-world applications where the flow conditions may vary and/or the manufactured shape may deviate from the CAD model. This led to the development of algorithms for shape optimization under uncertainties related to flow conditions and/or manufacturing imperfections. In the latter, the objective function to be optimized can be expressed as $\hat{F} = \hat{F}(\mathbf{c}, \mathbf{b}, F)$ to denote the dependency of \hat{F} on the stochastically varying environmental variables $\mathbf{c} \in \mathbb{R}^M$, the design vector $\mathbf{b} \in \mathbb{R}^N$ and the performance metric F .

Associated with any design under uncertainties is the process of Uncertainty Quantification (UQ) which quantifies the effect of the uncertain variables on the performance (F). In large-scale problems, Monte-Carlo Asmussen and Glyn (2007), Morokoff and Caflisch (1995) methods are prohibitively expensive UQ techniques.

A viable alternative is the Polynomial Chaos Expansion (PCE) Xiu and Karniadakis (2002), Eldred and Burkardt (2009). There are two ways to implement the PCE. In the intrusive PCE, every uncertainty affecting the flow model is introduced in the governing equations, new PDEs are derived and numerically solved. In the non-intrusive PCE (niPCE), the evaluation software is used as a black-box to compute the objective function values for some data-sets (determined by the Gauss integration formulas) of the uncertain variables.

In this work, the niPCE method is used together with an evolutionary algorithm to create a workflow for shape optimization under uncertainties. CAD-free approaches are utilized for shape deformations and a mesh morphing/smoothing tool, namely the Rigid Motion Mesh Morpher (R3M) Eleftheriou and Pierrot (2014), for the adaptation of the CFD mesh to the changed boundaries.

It is R3M and its corresponding smoother that generate the geometrical imperfections this paper is dealing with. Three applications are demonstrated, based on which the way of introducing geometrical imperfections is investigated. The first case deals with the optimization under geometrical imperfections of an S-bend duct, the second with a 2D manifold and the last with a two-element airfoil.

Design-Optimization Under Uncertainties

An Evolutionary Algorithm (EA), assisted by surrogate evaluation models or metamodels, is used for the optimization under uncertainties. In fact, this is the Metamodel-Assisted EA (MAEA) of the general purpose optimization platform EASY (Evolutionary Algorithms SYstem <http://velos0.lt.mech.ntua.gr/EASY>) which can handle single- or multi-objective, constrained or unconstrained problems. EASY handles three populations, namely μ parents, λ offspring and the elite set and applies evolution operators in conformity with binary or real encoding of the design vector (\mathbf{b}). For each offspring, the uncertainty of the function of interest F (such as drag, lift, losses, etc.) should be quantified. Since UQ using niPCE involves many calls to the CFD tool, a MAEA that uses low-cost surrogate evaluation models (radial basis functions networks) is the right method to reduce the computational cost.

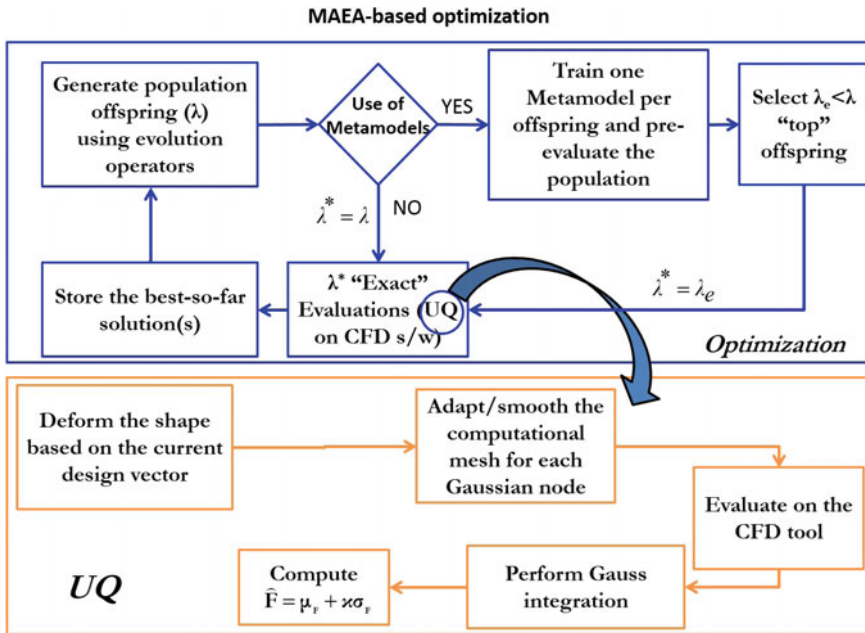


Fig. 1 Workflow for CFD-based shape optimization under manufacturing uncertainties leading to geometrical imperfections. The background optimization tool is a (μ, λ) EA, with μ parents and λ offspring in each generation

Local metamodels are on-line trained for each and every new individual generated during the evolution. For all but the first generations, metamodels (RBF networks in whatever follows) are used to pre-evaluate the offspring population by, practically, interpolating the objective function values of some of the previously evaluated individuals and indicate the most promising members to undergo CFD-based evaluation Karakasis and Giannakoglou (2006).

The optimization workflow in the case with manufacturing uncertainties is presented in Fig. 1. Topics such as the UQ using the niPCE, shape parameterization and mesh morphing are discussed below, in detail.

UQ Using Non-intrusive PCE

Let $F(\xi)$ be a function where ξ is a stochastic variable and $w(\xi)$ its probability density function (normal distribution). According to the PCE theory Xiu and Karniadakis (2002), F can be approximated by a linear combination of a finite subset of orthogonal polynomials $\Psi_i(\xi)$ (of degree i ; normalized Hermite polynomials)

$$F(\boldsymbol{\xi}) \approx \sum_{i=0}^q \alpha_i \Psi_i(\boldsymbol{\xi}) \quad (1)$$

with q being the chaos order, truncating Eq. 1. The first two statistical moments of F , i.e. its mean value and variance, can be written as

$$\mu_F = \int F(\boldsymbol{\xi}) w(\boldsymbol{\xi}) d\boldsymbol{\xi} = \alpha_0, \quad \sigma_F^2 = \int \left(F(\boldsymbol{\xi}) - \mu_F \right)^2 w(\boldsymbol{\xi}) d\boldsymbol{\xi} = \sqrt{\sum_{i=1}^q \alpha_i^2} \quad (2)$$

The PCE coefficients ($\alpha_i, i \in [0, q]$) result from the following integrations

$$\alpha_i = \int_{-\infty}^{\infty} n F(\boldsymbol{\xi}) \Psi_i(\boldsymbol{\xi}) w(\boldsymbol{\xi}) d\boldsymbol{\xi} \quad (3)$$

computed using Gauss Quadrature (GQ) Golub and Welsch (1969). To do so, the evaluation of the problem specific function is needed at a predefined number of the so-called Gaussian nodes.

After having computed the statistical moments of F through evaluations at the Gaussian nodes, the appropriate objective function(s) to be maximized or minimized can be computed. Either a multi-objective optimization problem, by seeking the Pareto front on the (μ_F, σ_F) plane, or a single-objective one, by concatenating the statistical moments into a single function, can be used. In this work, the objective function (\hat{F}) to be minimized is defined as

$$\hat{F} = \mu_F + \kappa \sigma_F \quad (4)$$

where κ is a user-defined (possibly signed) weight.

Shape Parameterization and Mesh Morphing

In this paper, without loss in generality, shape parameterization is based either on Radial Basis Functions (RBFs) or cages associated with a coarse mesh that control the CFD one through properly computed Harmonic Coordinates (HC) at the nodes of the latter. The coordinates of either the RBF centers or the HC cage knots constitute the design vector $\mathbf{b} \in \mathbb{R}^N$.

Radial Basis Function Model

K RBF centers are initially selected; these can either be a subset of the surface nodes or any set of points around the shape. In the applications shown in this paper, the RBF centers do not necessarily coincide with the surface nodes. The displacement $\Delta \mathbf{r}$ of any surface node, initially being at position \mathbf{r} , is given by

$$\Delta \mathbf{r} = \sum_{i=1}^K \mathbf{w}_i \phi(\|\mathbf{r}_{c,i} - \mathbf{r}\|) \quad (5)$$

where $\mathbf{r}_{c,i}$ is the initial position vector of the i -th RBF center, ϕ is the RBF activation function and \mathbf{w}_i are as many weights as the RBF centers, for each Cartesian direction. To compute the weights, Eq. 5 is applied at the K RBF centers (separately for each Cartesian coordinate) and the resulting linear systems are numerically solved.

The HC Two-Cage Model

Harmonic coordinates (HC), initially proposed for character articulation Joshi et al. (2007), use a topologically flexible structure called “cage” to control deformations of 2D or 3D domains. An HC-based technique that may control both shape deformations and adapt the CFD mesh to the new geometry has been proposed in Kapsoulis et al. (2016) by adopting a two-cage control mechanism. The two-cage model allows smooth adaptation of the CFD mesh by avoiding mesh quality degradation due to large boundary displacements. The cages are filled with a very coarse unstructured mesh and, by applying appropriate conditions and solving as many Laplace equations as the number of the (internal) cage control knots, the nodal HC values are computed. The HC are interpolated from the cage coarse mesh to the CFD mesh and, then, any CFD mesh deformation can be explicitly defined by the cage control knots displacements.

Rigid Motion Mesh Morpher

Though the RBF networks or the HC control cages could also undertake the adaptation of the CFD mesh to the updated geometry, CFD mesh adaptation is herein controlled by a separate mesh morpher and corresponding smoother (R3M: Rigid Motion Mesh Morpher Eleftheriou and Pierrot 2014). The reason is that the aforesaid smoother can also be used to generate shape variations by mimicking manufacturing imperfections. The R3M morpher is capable of displacing the internal mesh nodes by minimizing a given distortion metric by favouring rigidity in the critical directions of imminent distortion, being thus able to handle mesh anisotropies.

The computational mesh, including boundary nodes, is split into a number of overlapping stencils to be kept as rigid as possible. Let $\mathbf{u}_{i,s}$ be the ideal displacement of node i belonging to stencil ‘s’; this stands for the displacement of the node assuming a rigid motion of the stencil it belongs to (translation and rotation, without any change in shape and size). Within the optimization loop, such an ideal situation is not possible since the displacements of the boundary nodes are determined by the value-set of design variables controlled by the EA which do not necessarily conform with the desired rigidity. To use R3M only for adapting the CFD mesh to the new boundary which is not affected by uncertainties, it suffices to minimize

$$E_1 = \sum_s w_s \sum_{i \in s} \mu_{is} (\mathbf{u}_i - \mathbf{u}_{is})^2 \tag{6}$$

where \mathbf{u}_i is the displacement of each CFD mesh node, w_s is a weight determining the importance of each stencil and μ_{is} a weight associated with node i of stencil s , μ_{is} accounts for mesh anisotropy, by favoring rigidity in directions of imminent distortion. In 3D, if N_n is the number of the inner CFD mesh nodes, Eq. 6 has $3N_n$ unknowns and E_1 can be minimized in the least squares sense.

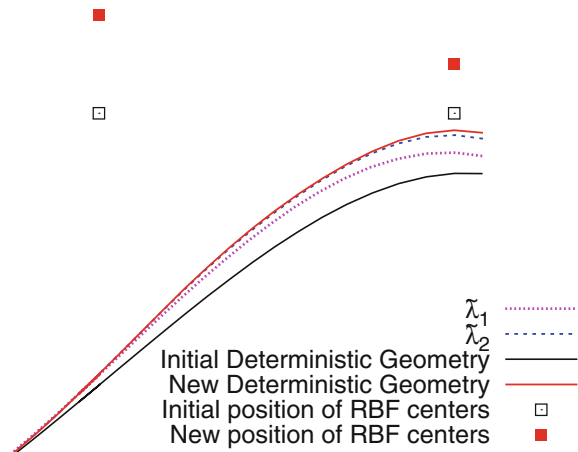
Over and above to mesh morphing, the same tool (R3M) can additionally be used to smooth the boundary. To do so, all boundary nodes belonging to patches controlled by the optimization algorithm are considered as “handles”. The position of these handles determines the shape of the boundary based on “spring theory”. In fact, each handle is connected with its underlying node with an ideal spring, the stiffness of which is controlled by a scalar coefficient $\tilde{\lambda}$. High $\tilde{\lambda}$ values cause smaller deviation from the wall shape, compared to the deterministic geometry, Fig. 2. The final positions of boundary and internal mesh nodes are, then, computed by minimizing

$$E = E_1 + \tilde{\lambda} \sum_{j \in \mathbb{H}} (\mathbf{u}_j - \mathbf{V}_j^t)^2 \tag{7}$$

where \mathbf{V}_j^t are the displacements corresponding to the deterministic geometry and \mathbb{H} the set of handles.

For the needs of this paper, the smoother (last term on the r.h.s. of Eq. 7) is used to create the stochastic variations in the boundary shape, by making the assumption that the uncertainty in the $\tilde{\lambda}$ value determines shape imperfections. Thus, for the known \mathbf{V}_j^t field (deterministic geometry resulting all from the EA-based search) the minimization of E (Eq. 7) provides a new CFD mesh with boundary different from the deterministic one, which is affected by the stochastically varying $\tilde{\lambda}$.

Fig. 2 Example of the effect of the $\tilde{\lambda}$ coefficient on mesh deformation (2D). For the displacements computed from the RBF model (red continuous line), the minimization of E (Eq. 7) determines the final nodal displacements (by considering imperfections) depending on the $\tilde{\lambda}$ values (blue/pink dashed lines for high/low values, respectively)



Applications

Optimization of an S-Bend Duct

This case deals with the shape optimization of an S-bend duct by considering geometrical imperfections. The optimization aims at minimizing \hat{F} , given by Eq. 4, where F stands for the total pressure losses between the duct inlet and outlet,

$$F = \frac{\int (p + \frac{1}{2}\rho\mathbf{u}^2)\mathbf{u} \cdot \mathbf{n}dS}{\int \mathbf{u} \cdot \mathbf{n}dS} \tag{8}$$

by deforming only the central curved part of the duct which is marked in red (Fig. 3). In Eq. 8, \mathbf{u} is the velocity vector, p is the pressure and \mathbf{n} is the outward unit normal vector at the boundaries of the flow domain.

The baseline 3D CFD mesh has been generated using CFD-GEOM CFD Research Corporation (1993) and consists of hexahedra close to the walls, a zone of prisms and tetrahedra everywhere else. The flow is laminar with the flow Reynolds number being equal to $Re = 550$. Uncertainty in $\tilde{\lambda}$ resulting in shape variations is assumed. In specific, $\tilde{\lambda}$ follows a normal distribution with mean value $\mu_{\tilde{\lambda}} = 0.017$ and standard deviation $\sigma_{\tilde{\lambda}} = 0.005$.

For each candidate solution generated during the optimization, UQ should be performed in order to obtain the mean value and standard deviation of F . The central curved part of the duct, which is free to deform, is controlled using an RBF model

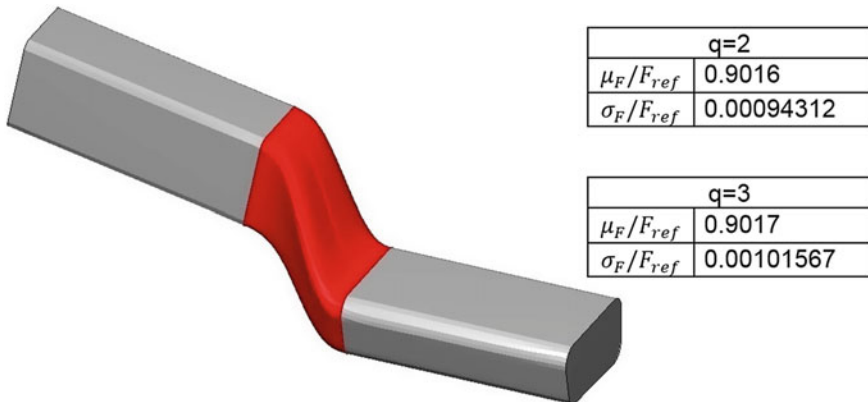


Fig. 3 S-bend duct. Left: Baseline geometry. Grey parts are kept fixed whereas the boundary marked in red is free to deform. Right: Computed mean value and standard deviation of total pressure losses (F) computed with the niPCE method for chaos order $q = 2$ and 3 for the optimized geometry. All tabulated quantities are normalized using the objective function of the baseline without uncertainties ($F_{ref} = 137.84$ Pa) as reference. Slight differences on the μ_F values depend on the integration using different Gaussian nodes (different GQ degree)

with $K = 24$ RBF centers. The coordinates of the latter (as in Eq. 5) are selected as design variables for the optimization workflow using EASY. An (8, 12) MAEA, with $\mu = 8$ parents and $\lambda = 12$ offspring, was used for the optimization; the termination criterion was set to 200 UQs. Metamodels were activated after the first 25 of them.

Optimization with chaos order equal to 2 was performed. This was a reasonable selection since, as shown in Fig. 3, the UQ using either $q = 2$ or $q = 3$ yields quite similar results; thus, $q = 2$, at the cost of 3 CFD runs per UQ, was selected. The optimized geometry yields an objective function (\hat{F}) value 9.6% lower than that of the baseline. The effect of the $\tilde{\lambda}$ value in the optimized S-bend geometry for chaos order $q = 2$ is shown in Fig. 4.

It is also worth comparing the results of the optimization of the S-bend duct under geometrical uncertainties with those resulting from a run without uncertainties. For this reason, the optimization without uncertainties has been performed, followed by the UQ on the optimized geometry for $q = 2$ and 3. Figure 5 presents the convergence histories of the optimizations with and without uncertainties. In Table 1, the mean value and the standard deviation of F computed for the optimized geometry (obtained from the run without considering uncertainties) are tabulated. The optimized geometry yields an objective function \hat{F} value which is by 10.4% lower than the baseline. All results have been normalized with the total pressure losses of the baseline geometry ($F_{ref} = 137.84$ Pa).

Comparing tables in Fig. 3 (optimization under uncertainties) and Table 1 (UQ in the optimized geometry without uncertainties), some differences can be noticed. The

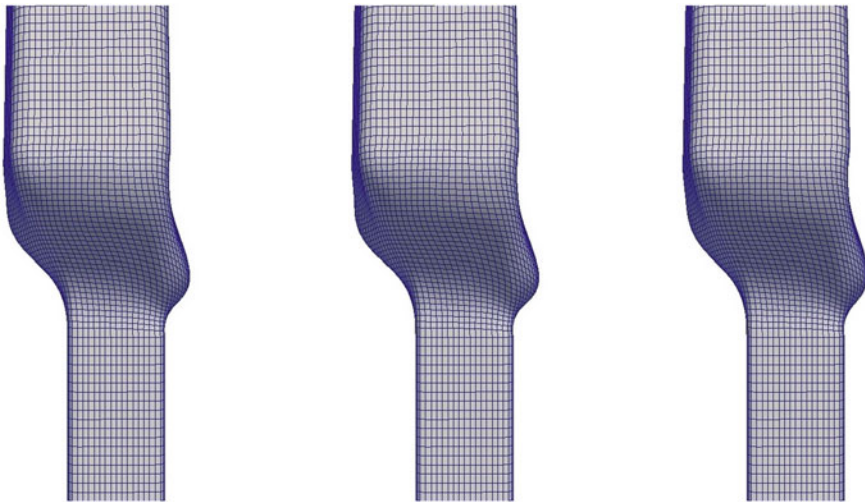


Fig. 4 S-bend duct. Effect of $\tilde{\lambda}$ to the optimized S-bend geometry for the three Gaussian nodes used for UQ with chaos order $q = 2$. Differences in the volume of the second and third geometry w.r.t. the first, caused by the variation in $\tilde{\lambda}$, are 0.22 and 0.47% respectively

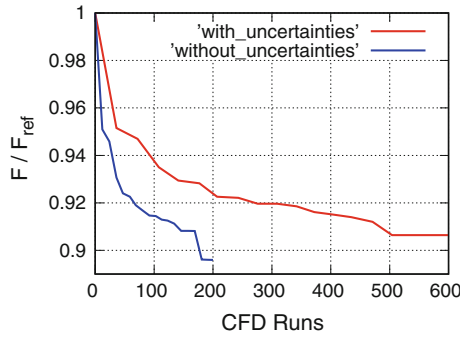


Fig. 5 S-bend duct. Convergence histories of the optimization with and without uncertainties

Table 1 S-bend duct. Mean value and standard deviation of total pressure losses computed with the niPCE method for $q = 2$ and $q = 3$ for the optimized geometry obtained from the run without considering uncertainties

Quantity	q = 2	q = 3
μ_F / F_{ref}	0.8961	0.8960
σ_F / F_{ref}	0.0024	0.0025

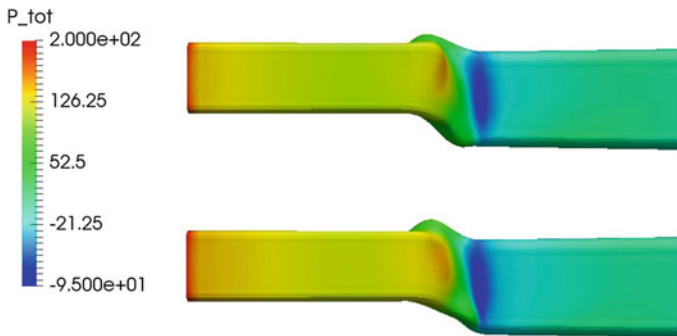


Fig. 6 S-bend duct case. Total pressure field in the optimized geometry resulted from the optimization with (top) and without (bottom) considering uncertainties

mean value of F in the latter run is lower than in the former whereas the standard deviation of F is three times higher. In Fig. 6, the total pressure field in the optimized geometries is presented. In the geometry generated by the optimization with uncertainties, the groove on the one side of the duct is smaller, which is probably the main reason for which this shape has lower standard deviation than the one generated from the optimization without uncertainties.

Optimization of a 2D Manifold

The second problem deals with the shape optimization of a 2D manifold with one inlet and three outlets, for minimum \hat{F} given by Eq. 4, with F being the total pressure losses across the duct (as in Eq. 8).

The baseline CFD mesh has approximately 140 K nodes and 70 K elements. An inlet velocity of $U_{in} = 0.3$ m/s leads to a laminar flow at $Re = 1300$. A single uncertainty in the coefficient $\tilde{\lambda}$ of the morpher is assumed, causing uncertainties in the manifold shape. It is assumed that $\tilde{\lambda}$ follows a normal distribution with mean value $\mu_{\tilde{\lambda}} = 0.3$ and standard deviation $\sigma_{\tilde{\lambda}} = 0.13$.

The baseline manifold shape, extruded in the third dimension for demonstration purposes, is shown in Fig. 7. Areas marked in red are free to deform. The velocity field in this geometry can be seen in Fig. 8 (left).

The manifold is parameterized using an HC control cage with 45 knots; 28 knots out of them are allowed to vary in both directions summing up to 56 design variables in total (Fig. 8; right). A (8, 12) MAEA was used and the metamodels were activated after the first 30 UQs. In the subsequent generations, all individuals were pre-evaluated on the metamodels and the top two of them in each generation were selected for CFD re-evaluations. After 300 UQs, a reduction in \hat{F} by $\sim 4\%$ was achieved.

The effect of chaos order on the optimization of the manifold duct is provided in Table 2. Chaos order equal to 2 appears to be a good compromise in terms of accuracy and computational cost. With a single uncertain variable, namely the $\tilde{\lambda}$ coefficient of the morpher and $q = 2$, three CFD evaluations per UQ are needed. The effect of the $\tilde{\lambda}$ value on the optimized geometry is presented in Fig. 10 and the convergence history in Fig. 9. All results are normalized with the total pressure losses of the baseline

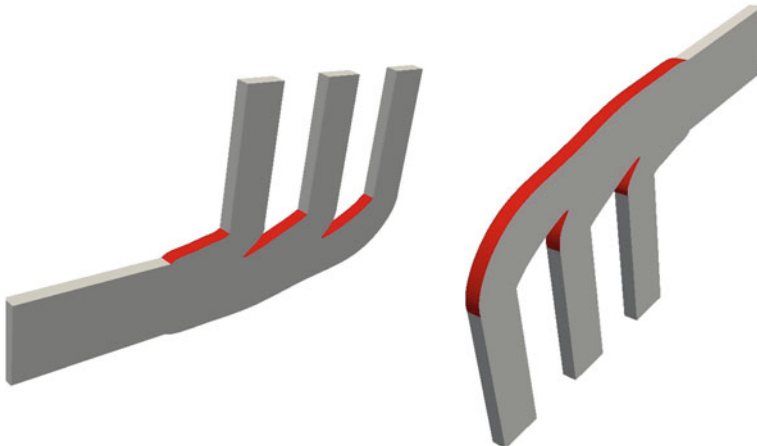


Fig. 7 Manifold case. Baseline geometry plotted in 3D for demonstration purposes. Deformable boundaries are marked in red

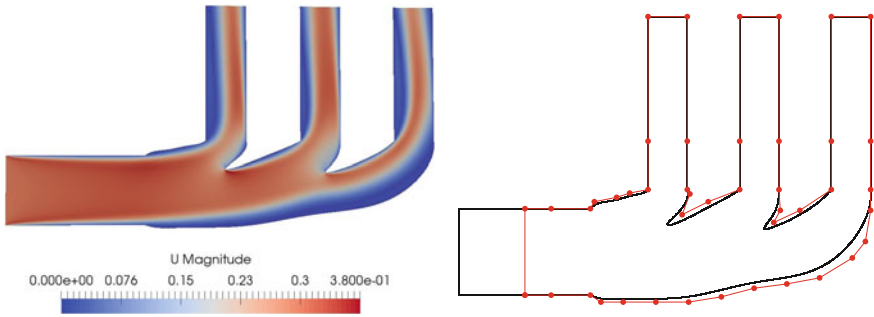
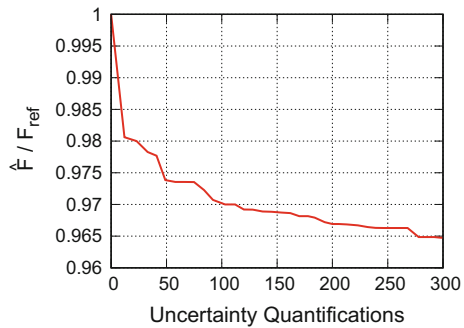


Fig. 8 Manifold case. Left: Velocity field in the baseline geometry; recirculation areas near the boundaries are the main reason for total pressure losses. Right: Baseline geometry (marked in black) and HC cage marked in red. Design variables correspond to the coordinates of the red nodes

Table 2 Manifold case. Mean value and standard deviation of F , for $q = 2$ and 3 , for the optimized geometry. Differences on μ_F depend on the integration with different Gaussian nodes

Quantity	$q = 2$	$q = 3$
μ_F / F_{ref}	0.9673	0.9674
σ_F / F_{ref}	0.000889	0.000887

Fig. 9 Manifold case. Convergence history of the optimization under uncertainties for $q = 2$. A reduction in \hat{F} by approximately 4% was achieved



geometry. Though only a small part of the manifold was free to deform, an important reduction of the \hat{F} was achieved.

Optimization of the Flap of a Two-Element Airfoil

The last case deals with the shape optimization of the flap of a two-element airfoil (Fig. 11, left), without changing the shape of the main body for maximum \hat{F} (given by Eq. 4 with $k = -1$). The performance metric F used herein is the lift coefficient C_L . The baseline CFD mesh consists of approximately 90K nodes and

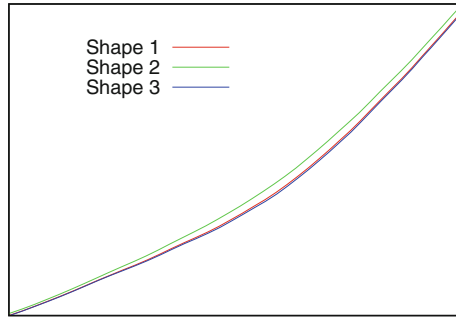


Fig. 10 Manifold case. Effect of the $\tilde{\lambda}$ value to the manifold for the three Gaussian nodes used for the UQ with $q = 2$. Close-up view at the deformable and the difference between the three geometries can be observed

Table 3 Two-element airfoil. Mean values and standard deviations of the uncertain variables. Normal distribution for all of them is assumed

Uncertain variable	μ	σ
$\tilde{\lambda}$	0.10	0.03
$\Delta x / chord_{flap}$	0.0067	0.0033
$\Delta y / chord_{flap}$	-0.0033	0.0023

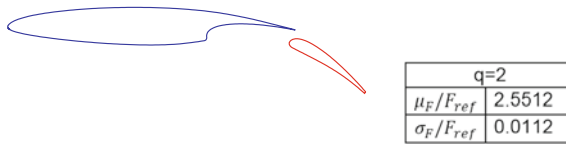


Fig. 11 Two-element airfoil. Left: Baseline geometry of the main body and flap. Right: Mean value and standard deviation of F for $q = 2$. The lift coefficient for the baseline geometry without considering uncertainties is $C_L = 2.5465$

155 K elements. The flow is incompressible and turbulent with freestream Mach number $M_\infty = 0.147$, Reynolds number based on the chord $Re_c = 4.23 \cdot 10^6$ and zero freestream flow angle. The Spalart–Allmaras turbulence model Spalart and Allmaras (1994) is used. In this case, three uncertain variables, that all follow normal distributions, were assumed. Uncertain variables are the $\tilde{\lambda}$ coefficient of the morpher and the flap positioning (Δx , Δy) w.r.t. the airfoil main body. The mean value and standard deviation of the uncertain variables are tabulated in Table 3.

The outcome of the UQ for $q = 2$ is demonstrated in Fig. 11 along with the main body of the airfoil, which is kept fixed, and the baseline geometry of the flap. The UQ with three variables and $q = 2$ requires 27 CFD runs to compute the mean value and standard deviation of the lift coefficient.

The flap is parameterized using HC cages. The control cage consists of 17 knots summing up to 34 design variables. The main body of the airfoil is kept fixed whereas the flap is allowed to deform. For the flap, the leading and trailing edges are not allowed to move. An increase in \hat{F} by $\sim 2\%$ was achieved leading to $\mu_F = 2.6001$ and $\sigma_F = 9.27 \cdot 10^{-3}$.

It can be noticed that the mean value of the lift coefficient of the optimized geometry is higher than that of in the baseline geometry whereas the standard deviation is lower. Thus, the optimized geometry (Fig. 13) operates more efficiently in a range of operating points. In Fig. 12, the Mach number for the baseline and the optimized shape is demonstrated. In the optimized geometry, the Mach number along the suction side is higher which is the reason of the increased lift coefficient.

The importance of using the low-cost surrogate models that EASY implements, is crucial in this case since for each candidate solution the UQ requires 27 CFD runs.

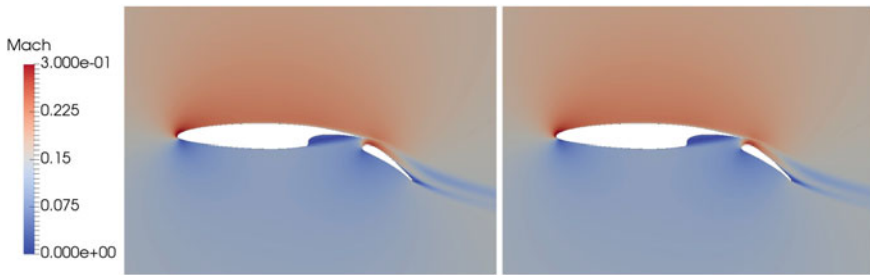
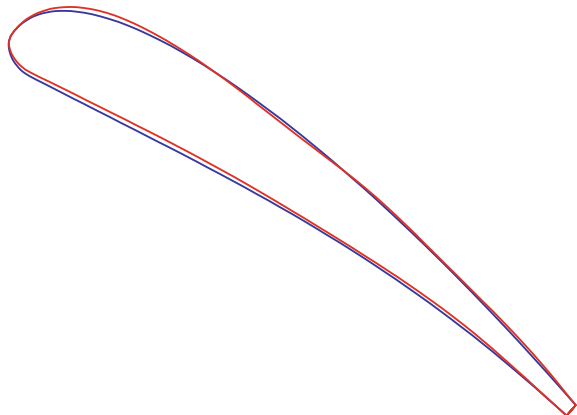


Fig. 12 Two-element airfoil. Mach number contours around the baseline (left) and the optimized (right) flap geometry. The optimized geometry has been evaluated for the mean value of all uncertain variables

Fig. 13 Two-element airfoil. Close-up view on the flap (baseline in blue; optimized in red). The curvature of the mean camber line is increasing, to maximize the lift coefficient



Closure

This paper presented a way to implement geometrical (manufacturing) imperfections during the aerodynamic shape optimization under uncertainties. This is done through the Rigid Motion Mesh Morpher (R3M) and its corresponding smoother. Uncertainty quantification was based on the non-intrusive PCE and the optimization was carried out by a metamodel–assisted EA. The use of metamodels was beneficial since it led to a reduced number of flow solutions which, in the case of UQ (with several uncertain variables), involves several calls to the CFD s/w. All these tools have been put in the form of an automated workflow for performing optimization under manufacturing uncertainties. Three applications in internal and external aerodynamics have been presented, with up to three uncertain variables related to the shapes themselves.

Acknowledgements The first author acknowledges the support from the People Programme (ITN Marie Curie Actions) of the European Union’s H2020 Framework Programme (MSCA-ITN-2014-ETN) under REA Grant Agreement no. 642959 (IODA project). The first author is an IODA Early Stage Researcher and PhD student at NTUA.

References

- Asmussen S, Glynn P (2007) Stochastic simulation: algorithms and analysis. Stochastic Modelling and Applied Probability Springer, New York
- CFD Research Corporation (1993) CFD-ACE Theory Manual. <https://books.google.de/books?id=FinRGwAACAAJ>
- Eldred MS, Burkardt J (2009) Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In: 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition
- Eleftheriou GS, Pierrot G (2014) Rigid motion mesh morpher: a novel approach for mesh deformation. In: OPT-i, international conference on engineering and applied sciences optimization, Kos Island, Greece
- Golub G, Welsch J (1969) Calculation of Gauss quadrature rules. *Mathe. Comput.* 106(23):221–230
- Joshi P, Meyer M, DeRose T, Green B, Sanocki T (2007) Harmonic coordinates for character articulation. *ACM Trans Graph* 26(3), Art. No. 71
- Kapsoulis DH, Tsiakas KT, Asouti VG, Giannakoglou KC, (2016) The use of kernel PCA in evolutionary optimization for computationally demanding engineering applications. (2016) IEEE symposium series on computational intelligence (IEEE SSCI 2016). Greece, Athens
- Karakasis MK, Giannakoglou KC (2006) On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Eng Optim* 38(8):941–957
- Morokoff W, Caflisch R (1995) Quasi-monte carlo integration. *J Comput Phys* 122(2):218–230
- Spalart PR, Allmaras SR (1994) A one-equation turbulence model for aerodynamic flows. *La Recherche Aeronautique* 1:5–21
- The EASY (Evolutionary Algorithms SYstem) software. <http://velos0.ltt.mech.ntua.gr/EASY>
- Xiu D, Karniadakis G (2002) The Wiener - Askey polynomial chaos for stochastic differential equations. *SIAM J Sci Computing* 24(2):619–644

Multiobjective Optimisation of Aircraft Trajectories Under Wind Uncertainty Using GPU Parallelism and Genetic Algorithms



Daniel González-Arribas, Manuel Sanjurjo-Rivo
and Manuel Soler

Abstract The future Air Traffic Management (ATM) system will feature trajectory-centric procedures that give airspace users greater flexibility in trajectory planning. However, uncertainty generates major challenges for the successful implementation of the future ATM paradigm, with meteorological uncertainty representing one of the most impactful sources. In this work, we address optimized flight planning taking into account wind uncertainty, which we model with meteorological Ensemble Prediction System forecasts. We develop and implement a Parallel Probabilistic Trajectory Prediction system on a GPGPU framework in order to simulate multiple flight plans under multiple meteorological scenarios in parallel. We then use it to solve multiobjective flight planning problems with the NSGA-II genetic algorithm, which we also partially parallelize. Results prove that the combined platform has high computational performance and is able to efficiently compute tradeoffs between fuel burn, flight duration and trajectory predictability within a few seconds, therefore constituting a useful tool for pre-tactical flight planning.

Introduction

The current Air Traffic Management (ATM) system has significant limitations and sources of inefficiency, which has led the European Union (through SESAR¹), the US (through NextGen²), Japan and other countries to develop and implement a new

¹<http://www.sesarju.eu/vision>

²<https://www.faa.gov/nextgen/>

D. González-Arribas (✉) · M. Sanjurjo-Rivo · M. Soler
Department of Bioengineering and Aerospace Engineering,
Universidad Carlos III de Madrid, 28911 Leganés, Spain
e-mail: dangonza@ing.uc3m.es

M. Sanjurjo-Rivo
e-mail: msanjurj@ing.uc3m.es

M. Soler
e-mail: masolera@ing.uc3m.es

ATM paradigm in order to attain improvements in airspace capacity, safety, efficiency and environmental impact. The rigid airspace structures of today will be replaced by a trajectory-centric concept (called Trajectory-Based Operations or TBO) where airspace users will choose their trajectories with greater freedom. Flight plans will be produced in a collaborative and layered process that ensures that the airline business interests are met while respecting capacity or environmental constraints.

Meteorology has a significant influence in both flight efficiency and airspace performance; it is, therefore, essential to take weather forecasts into account when building flight plans. Methods for finding optimal trajectories using wind forecasts can be grouped in analytical optimal control-based techniques (Jardin and Bryson 2001, 2012; Sridhar et al. 2011; Marchidan and Bakolas 2016), dynamic programming-like methods (Girardet et al. 2014), and direct methods for numerical optimal control (Bonami et al. 2013; Soler et al. 2015; González Arribas 2016; González-Arribas 2017). With the exception of our previous work (González Arribas 2016; González-Arribas 2017), there has been little consideration of uncertainty and its impact on the predictability of the trajectory within a flight planning context (with some exceptions, such as Cheung 2015).

Understanding and managing uncertainty, nevertheless, is essential for increasing ATM predictability, which is in turn necessary in order to realize the full benefits of the TBO concept (see Cook and Rivas 2016, Chap. 4). Meteorological uncertainty is one of the main sources of trajectory uncertainty; therefore, it is becoming increasingly critical to consider it in a flight planning context.

In a Numerical Weather Prediction (NWP) context, meteorological uncertainty arises naturally from incomplete knowledge of the state of the atmosphere, model error in physical parametrizations, computational limitations and nonlinear, sometimes chaotic, dynamics. In response to this challenge, NWP researchers and practitioners developed the Ensemble Prediction System (EPS) concept. An EPS is composed by 10 to 100 individual forecasts, called “ensemble members”, each one running with strategically perturbed initial conditions or parameters, thus providing a probabilistic estimation of the future state of the atmosphere. They have become a fundamental tool for NWP centers. See Bauer et al. (2015) for more details.

EPS forecasts are starting to be employed by the ATM research community (Steiner et al. 2010; Cheung et al. 2014; Cheung 2015); following this trend, our previous work in González Arribas (2016) and González-Arribas (2017) used them for flight planning purposes with a methodology based on optimal control and direct collocation. However, while this methodology is effective and relatively fast, it can only generate one solution at a time. Exploring the trade-offs between fuel efficiency, flight duration and predictability, which demand the computation of multiple solutions, is a slower task. In addition, the calculated trajectories are locally optimal, which does not necessarily imply global optimality. Finally, a good initial guess is needed in order to achieve good performance.

Therefore, there is still a need for a fast and scalable trajectory optimization system for solving flight planning problems with uncertainty. The goal of this paper is to identify and build a methodology that tackles the computational challenges posed by this type of problem.

The current work introduces a Parallel Probabilistic Trajectory Predictor (PPTP) system that can propagate multiple flight plans under different meteorological scenarios in parallel through General-purpose computing on Graphics Processing Units (GPGPU). It is shown that the current capabilities of graphics processing hardware are well-suited for the solution of uncertain problems that can be described by a discrete set of scenarios. We pair the PPTP with a genetic algorithm, NSGA-II (Deb et al. 2002), in order to create a system that can generate a solution portfolio where a user can select an optimized flight plan that best matches her preferences regarding fuel burn, flight duration and trajectory predictability.

This paper is structured as follows. We start by describing the PPTP in Section “Parallel Probabilistic Trajectory Predictor”; then, we describe the multiobjective optimization methodology in Section “Flight Plan Optimization”. We then apply the complete system to a case study in Section “Results” in order to study the performance of the system. Finally, we summarize our findings and discuss future work in Section “Conclusions”.

Parallel Probabilistic Trajectory Predictor

In this section, we describe the GPU-based Parallel Probabilistic Trajectory Predictor (PPTP) platform that will be employed in this study. This tool can be used for efficient simulation of en-route flight plans under multiple meteorological scenarios.

Modeling

Since the impact of wind uncertainty is cumulative and the cruise phase constitutes most of a medium-haul or long-haul flight, we consider the cruise phase of a commercial flight. We also consider a constant flight level for demonstration purposes, but we plan to incorporate variable flight levels in the future.

We model the Earth as an ellipsoid (as in the WGS-84 model), denoting the radii of curvature of ellipsoid meridian and prime vertical by R_M and R_N respectively. The meteorological parameters (wind (w_x, w_y) , temperature T , and geopotential height z) are drawn from an EPS forecast by interpolating the two closest pressure levels to the corresponding flight level³. Density is computed at each point with the aid of the ideal gas law:

$$\rho = \frac{P}{R_s T}$$

where $R_s = 287.058 \text{ J}/(\text{Kg}\cdot\text{K})$ is the specific gas constant of the air and P is the pressure corresponding to the specific flight level.

³Note that a flight level corresponds to barometric altitude, i.e. a constant pressure level

We will employ a 3-DoF point-mass model of the aircraft, as it is widely done in ATM research. We neglect turn dynamics; in smooth long-range trajectories, the bank angle is very small and removing turn dynamics does not significantly degrade accuracy. We denote latitude by ϕ , longitude by λ , the true airspeed by v and the aircraft mass by m . These four variables constitute the state variables of the dynamical system under the specified assumptions, and they evolve according to the differential equation:

$$\frac{d}{dt} \begin{bmatrix} \phi \\ \lambda \\ v \\ m \end{bmatrix} = \begin{bmatrix} (R_N + z)^{-1}(v \cos(\chi) + w_x(\phi, \lambda, t)) \\ (R_M + z)^{-1} \cos^{-1}(\phi)(v \sin(\chi) + w_y(\phi, \lambda, t)) \\ (Tr - D(m, v))/m \\ -\eta(v, T)Tr \end{bmatrix} \quad (1)$$

where Tr denotes the thrust force, χ denotes the heading angle (measured with respect to the geographic North), D represents the drag force and η represents the thrust-specific fuel consumption. The drag is computed as a function of the lift L according to the constant-altitude assumption $L = mg$. Both the aerodynamic forces D and L and the fuel consumption η are computed according to the BADA 4 Aircraft Performance Model described in Gallo et al. (2006).

It is also useful to introduce the ground speed v_G and the course ψ , which are related to the wind, the true airspeed and the heading angle by (see Fig. 1 for illustration of Eq. 3):

$$\begin{aligned} v_G \cos \psi &= v \cos(\chi) + w_x(\phi, \lambda, t) \\ v_G \sin \psi &= v \sin(\chi) + w_y(\phi, \lambda, t) \end{aligned} \quad (2)$$

Note that the course (and not the heading) is what determines the direction of the movement of the aircraft, and thus it is the variable that is controlled in order to follow a flight plan. Given a value of the wind (w_x, w_y), an airspeed value v and a course ψ , the ground speed can be computed with the following formulas:

$$\begin{aligned} w_{alongtrack} &= w_y \cos \psi + w_x \sin \psi \\ w_{crosswind} &= -w_y \sin \psi + w_x \cos \psi \\ v_{proj} &= \sqrt{v^2 - w_{crosswind}^2} \\ v_G &= v_{proj} + w_{alongtrack} \end{aligned} \quad (3)$$

Fig. 1 Relationship between airspeed, groundspeed, wind, heading and course

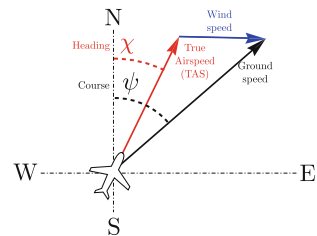
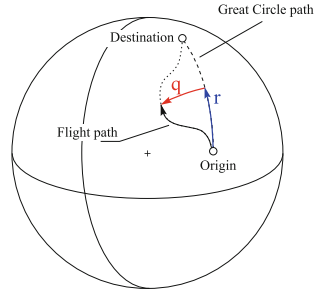


Fig. 2 Geographical coordinate system employed for the encoding of the flight path



Flight Plan Encoding

At a constant flight level, a free routing flight plan is composed by a lateral path and an airspeed profile. Here, we describe here the scheme by which we encode both parts of the flight plan; later on, these encoding variables will also serve as decision variables of the optimization problem.

In order to describe the lateral path, we introduce a coordinate system where the position is defined by the value of the (r, q) coordinates. Fig. 2 illustrates the concept behind this coordinate system. Let $r_0 \in \mathbb{R}^3$ be the position on the unitary sphere defined by the latitude and longitude of the point of origin (and r_f analogously by the destination point). We define the following unitary vectors:

$$\hat{r}_{ax} = \frac{r_0 \times (r_f - r_0)}{\|r_0 \times (r_f - r_0)\|}$$

$$\hat{q}_{ax} = \frac{r_f - r_0}{\|r_f - r_0\|}$$

Let θ be the angle such that the origin is translated to the destination by a rotation around \hat{r}_{ax} of angle θ . Then, we can describe any position on the sphere by the following rotations:

1. A rotation of the origin around \hat{r}_{ax} by an angle of $\theta(1 + r)/2$, where r is a scalar coordinate.
2. A rotation of the obtained point around \hat{q}_{ax} by an angle of q , where q is a scalar coordinate.

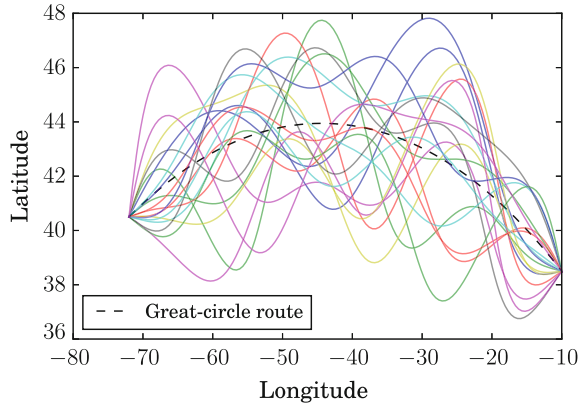
These rotations can be efficiently computed with a simplified version of Rodrigues' rotation formula⁴. Thus, r goes from -1 to 1 as the aircraft moves from the

⁴According to the Rodrigues' rotation formula, a rotation of vector v by an angle θ around unit vector k can be computed as:

$$v_{rot} = v \cos \theta + (k \times v) \sin \theta + k(k \cdot v)(1 - \cos \theta)$$

The last term vanishes if v and k are orthogonal, as it is the case in our application.

Fig. 3 Example lateral paths



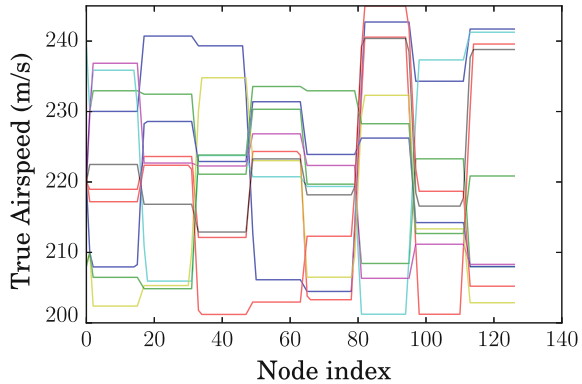
origin to the destination while q describes the deviation from the orthodromic path in a perpendicular direction. We define a valid lateral path as a smooth function $q(r) \in C^\infty([-1, 1])$ such that $q(-1) = q(1) = 0$ (i.e. vanishes in the boundaries, in order for the path to reach the origin and destination). Given a degree of the expansion n_{ec} and a vector $l \in \mathbb{R}^{n_{ec}}$, we can build one such function by the expansion:

$$q_l(r) = \sum_{k=0}^{n_{ec}-1} l_k \cos\left(\frac{\pi}{2}(k + (k + 1)r)\right) \tag{4}$$

Thus, given an origin and destination defining the coordinate system, the expansion coefficients l_k completely determine a lateral path and we use them to encode the lateral path information. Figure 3 displays some lateral paths built from randomly-generated vectors l . In practice, we will restrict their values so that $\|q_l(r)\|_\infty = \max_{r \in [-1, 1]} |q_l(r)| \leq \pi/2$. Once we have built a path, we will discretize it into n_{nodes} points and compute the associated values of the longitude and latitude; we assume that the aircraft will then fly a loxodromic (constant course) path between each pair of points.

We use a simpler scheme for the encoding of the airspeed profile. We subdivide the n_{nodes} into n_{ts} segments of constant airspeed, with the airspeed transitioning between segments at a rate that is consistent with thrust limitations. The airspeed profile is then represented by a vector of coefficients $\tilde{v} \in \mathbb{R}^{n_{ts}}$ $\tilde{v}_j \in \{v_{\min}, v_{\max}\}$, $j \in \{0, \dots, n_{ts} - 1\}$, where v_{\min} and v_{\max} are derived from the flight envelope of the aircraft. Some randomly generated examples of airspeed profiles can be seen in Fig. 4.

Fig. 4 Example airspeed profiles



Implementation

We implement the PPTP in the CUDA platform (NVIDIA Corporation 2010), which allows the usage of NVIDIA Graphics Processing Units (GPU) for general-purpose computation. A programmer can use CUDA to partition a computational problem into tasks that can be executed in parallel for automatic scalability.

A CUDA program is called a *kernel*. A kernel is executed in parallel *threads*, which are grouped into *blocks*. When launching the kernel, the programmer needs to specify a *grid*, which defines the number and configuration of the blocks. At this point, the blocks are distributed to the Streaming Multiprocessors (SM) of the GPU, which partition the blocks into groups of 32 threads called *warps* and starts executing them in its CUDA cores (the individual computing units that form a SM).

In order to maximize efficiency, the programmer needs to ensure that the device has maximum utilization (by generating enough tasks to keep all the CUDA cores busy, while preventing branching that would keep some threads inactive) and optimize device memory accesses (for example, by ensuring that threads access contiguous memory addresses and the access is strided in order to minimize the number of required memory loads).

The PPTP is based on PyCUDA (Klöckner et al. 2012). When the problem is initialized, PPTP reads the CUDA source files and formats them with the specific parameters of the aircraft and the problem using a templating engine in a meta-programming fashion. PPTP then searches and loads the required weather forecasts from a disk cache; if it does not find them, PPTP fetches them from the TIGGE dataset (Buizza 2015) using the ECMWF⁵ API. Finally, it initializes the GPU, transfers the weather data to the GPU memory, initializes the arrays and compiles the formatted source code.

At this stage, PPTP is ready for simulation, which is composed by two parts: a preprocessing step and a main loop. Each simulation takes n_{fp} flight plans (which we will denote with the subindex i) and will run each flight plan for every weather

⁵European Centre for Medium-Range Weather Forecasts

scenario $j \in \{0, \dots, N - 1\}$. Note that the weather scenario j corresponds to the j -th member of the EPS forecast. Both the preprocessing step and the main loop are performed in parallel, with a block per flight plan; however, a block in the preprocessing step contains a thread per path point, while a block in the main loop contains a thread per weather scenario.

In the preprocessing step, PPTP computes the latitude-longitude paths from the expansion coefficients l^i associated to the flight plan i and discretizes it into n_{nodes} latitude-longitude path points or nodes, which define $n_{nodes} - 1$ segments. Then, it computes the loxodromic distances between nodes as well as the course of the aircraft between the nodes. Finally, it generates the airspeed profiles from the airspeed vectors \tilde{v}^i . In the main loop, two differential equations are integrated in the node grid. In the first, we use distance flown s as the independent variable and we integrate flyover time $t_{i,j}(s)$, which represents the time at which the aircraft is at position s along the route in the i -th flight plan and the j -th weather scenario:

$$\frac{dt_{i,j}}{ds_i} = \frac{1}{v_{G,i,j}} \quad (5)$$

The second differential equation describes the evolution of the mass:

$$\frac{dm_{i,j}}{dt_{i,j}} = -\eta(v_i, T)Tr_{i,j}^{req}(m) \quad (6)$$

where $Tr_{i,j}^{req}$ is the thrust that is required in order to match the airspeed profile. Both equations are integrated with Heun's method:

$$\begin{aligned} y'(t) &= f(t, y(t)) \\ \tilde{y}_{i+1} &= y_i + hf(t_i, y_i) \\ y_{i+1} &= y_i + \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, \tilde{y}_{i+1})] \end{aligned} \quad (7)$$

where the step size h is determined by the spacing between grid nodes. While both equations are integrated in lockstep, the time evolution (independent of mass) is integrated first, resulting in a more accurate integration of the mass.

The weather variables (wind for the calculation of the groundspeed, temperature for the fuel burn and aerodynamic calculations) are implemented as "texture fetch operations", using the texture units on the GPU for economic access and interpolation of the meteorological variables.

All the computations inside the GPU are performed in single-precision (32-bit) floating point arithmetic, which makes the computations faster than under double-precision (64-bit) floating point numbers. The accuracy of the solution is not heavily damaged by the usage of 32-bit arithmetic since we don't rely on numerically unstable or ill-conditioned operations.

Flight Plan Optimization

Optimization Objectives

Once the PPTP simulates a set of flight plans, it stores fuel burn $\Delta m = m_f - m_0$ and flight duration $\Delta t = t_f - t_0$ for each flight plan and ensemble member. It then computes, for each flight plan i , the average fuel burn, average flight time and the spread in arrival times:

$$\begin{aligned}\bar{\Delta m}_i &= \frac{1}{N} \sum_{k=1}^N \Delta m_{i,j} \\ \bar{\Delta t}_i &= \frac{1}{N} \sum_{k=1}^N \Delta t_{i,j} \\ w_i &= \max_j \Delta t_{i,j} - \min_j \Delta t_{i,j}\end{aligned}\tag{8}$$

where the subindex i, j denotes the value of a variable for the flight plan i and the ensemble member j . The spread in arrival times w_i is the difference between the earliest and the latest time of arrival under flight plan i and is used to characterize the predictability of the trajectory. These three metrics can be used to define a multiobjective optimization problem: finding the flight plans (l, \tilde{v}) that minimize $\bar{\Delta m}$, $\bar{\Delta t}$ and w .

Solution Approach

We choose to solve the problem with the NSGA-II algorithm (Deb et al. 2002), a widely-employed elitist multiobjective genetic algorithm. Each iteration in the NSGA-II main loop is composed by the following procedures:

1. Non-dominated sort: the individuals of the population are placed in ordered fronts such that an individual in a front is dominated by individuals in the previous front (except for the first front, which contains the individuals that are not dominated in all objectives by any other individual). This is the main selection criterion in NSGA-II.
2. Crowding distance computation: a measure of the density of solutions surrounding a particular individual in the population is calculated. This provides the secondary selection rule in NSGA-II, employed for individuals that belong to the same front.
3. Selection of the parent population: the best half of the population according to the described selection criteria is elected as potential parents.
4. Generation of the offspring population: a binary tournament selection with the described selection rule is employed to select parents from the pool of potential parents. The offspring are generated by using simulated binary crossover and

polynomial mutation. Finally, the new generation is comprised by the parents and the offspring.

We implement the non-dimensional sort and the generation of the offspring population in CUDA. The remaining procedures are implemented at the CPU side. For the non-dimensional sort, we compute the domination matrix S and the vector n that accounts for the amount of individuals that dominate a given individual in parallel; when filling each front, we update the vector n in parallel as well.

Results

Description of the Case Study

We consider a scenario based in the one described in González Arribas (2016). An A330⁶ flies from the vertical of New York to the vertical of Lisbon at flight level FL380 with initial mass 200000 Kg. We employ a 50-member EPS forecast with 6-hours lead time produced by the ECMWF and hosted at the TIGGE dataset (Buizza 2015). The computational parameters are reproduced in Table 1, and the parameters of the NSGA-II algorithm are set to their default values in Deb et al. (2002).

Computational Performance

With the specified computational parameters, simulating a generation takes an average of 9.4 ms, which implies that the raw throughput of the PPTP is 106 generations per second, 11,000 flight plans per second or around 550,000 trajectories per second. The NSGA-II algorithm adds an overhead of around 1.4 ms per generation, which reduces throughput when using the complete algorithm to around 91 generations per second.

Optimization Results

In first place, we run the algorithm considering only average flight time and fuel burn as objectives. Figure 5 illustrates the optimized flight paths; it can be observed that the Eastbound trajectory takes advantage of the jet stream while the Westbound trajectory avoids it. Figure 6 displays the Pareto front after several generations, showing that the front has almost converged at 800 generations (~9 sec of computation time, which

⁶We consider an A330-231, with BADA 4 code A330-321

Table 1 Parameter values

Parameter	Value	Description
n_{ec}	6	# of lateral path expansion coeffs.
n_{ts}	8	# of constant airspeed legs
n_{nodes}	128	# of nodes
n_{fp}	104	# of flight plans per generation
N	50	# of ensemble members

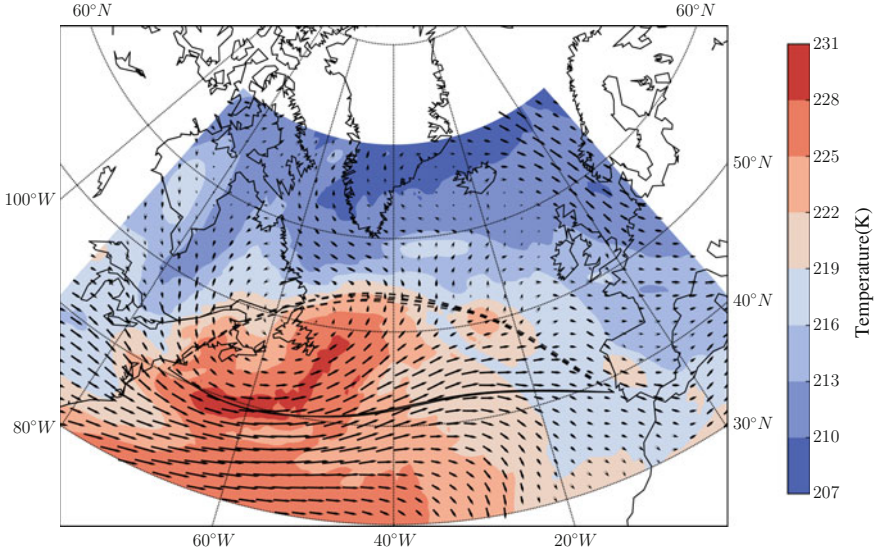


Fig. 5 Routes from origin to destination (solid black) and back (dashed black)

Fig. 6 Tradeoff between objectives Δm and Δt

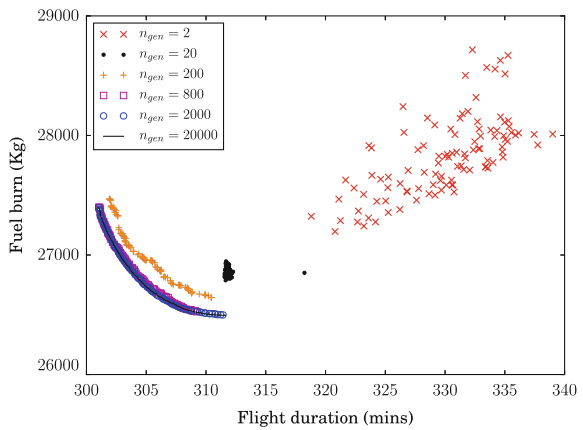


Fig. 7 Tradeoff between objectives $\Delta\bar{m}$ and $\bar{\Delta}t$

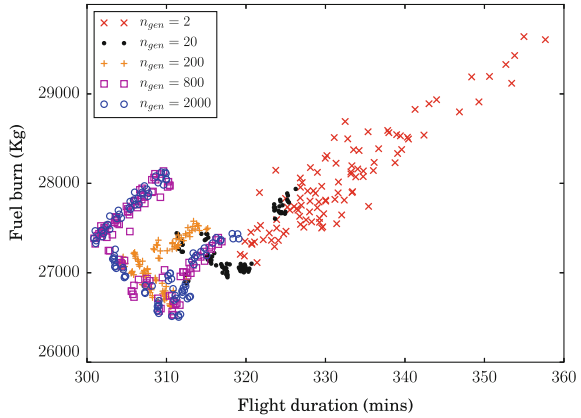
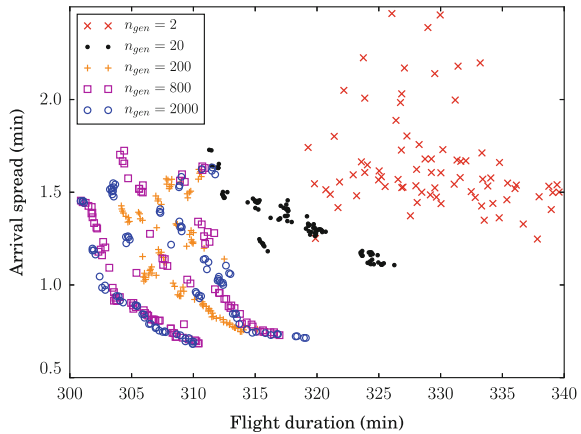


Fig. 8 Tradeoff objectives $\bar{\Delta}t$ and w



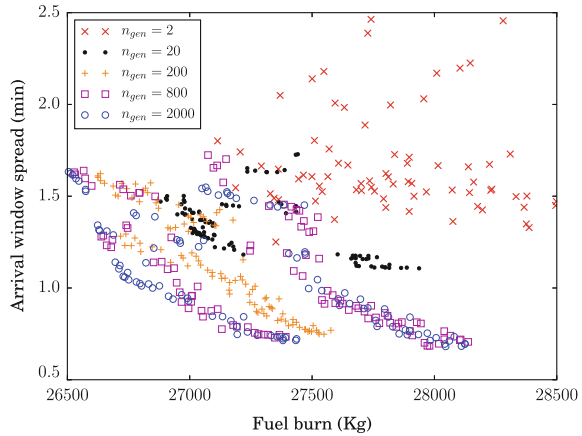
could be reduced to a fraction of that with a coarser grid and additional optimization), since it overlaps with the front at 20000 generations.

In second place, we add a predictability objective by considering the spread in the arrivals. Figures 7, 8, and 9 illustrate the objective values of the population after 2, 20, 200, 800, and 2000 iterations. It can be seen that some sections of the 3D Pareto front (such as the $\Delta\bar{m} - \bar{\Delta}t$ front) are already close to convergence after 200 iterations (~2 sec).

Conclusions

We have presented a scalable trajectory simulator under uncertainty that meets the necessary performance requirements for fast flight plan optimization under uncertainty and shown that modeling uncertainty through discrete scenarios (as it is done

Fig. 9 Tradeoff objectives Δm and w



in EPS forecasts) leads to problems that can be efficiently addressed through GPGPU techniques.

Our future work on this system will feature its extension to full 4D trajectory optimization problems by adding a vertical profile with variable altitude. We will incorporate other genetic and evolutionary algorithms in order to compare their performance. Finally, additional uncertainty sources (such as uncertainty in initial mass or uncertainty in aerodynamic parameters) will be considered.

References

Bauer P, Thorpe A, Brunet G (2015) The quiet revolution of numerical weather prediction. *Nature* 525(7567):47–55

Bonami P, Olivares A, Soler M, Staffetti E (2013) Multiphase mixed-integer optimal control approach to aircraft trajectory optimization. *J Guidance Control Dyn* 36(5):1267–1277

Buizza R (2015) The TIGGE global, medium-range ensembles. Technical Report 739, ECMWF

Cheung J, Brenguier J-L, Heijstek J, Marsman A, Wells H (2014) Sensitivity of flight durations to uncertainties in numerical weather prediction. In: *SIDs 2014 - Proceedings of the SESAR innovation days*

Cheung J, Hally A, Heijstek J, Marsman A, Brenguier J-L (2015) Recommendations on trajectory selection in flight planning based on weather uncertainty. In: *SIDs 2015—proceedings of the SESAR innovation days*

Cook A, Rivas D (2016) *Complexity science in air traffic management*. Taylor & Francis

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolutionary Comput.* 6(2):182–197

Gallo E, Navarro F, Nuic A, Iagaru M (2006) Advanced aircraft performance modeling for ATM: BADA 4.0 results. In: *2006 IEEE/AIAA 25th Digital Avionics Systems Conference* pp 1–12

Girardet B, Lapasset L, Delahaye D, Rabut C (2014) Wind-optimal path planning: application to aircraft trajectories. In: *2014 13th international conference on control automation robotics and vision (ICARCV)*, pp 1403–1408. IEEE

- González Arribas D, Soler M, Sanjurjo Rivo M (2016) Wind-based robust trajectory optimization using meteorological ensemble probabilistic forecasts. In: 6th SESAR innovation days eurocontrol
- González-Arribas D, Soler M, Sanjurjo-Rivo M (2017) Robust aircraft trajectory planning under wind uncertainty using optimal control. *J. Guidance Control Dyn* 1–16
- Jardin MR, Bryson AE (2001) Neighboring optimal aircraft guidance in winds. *J Guidance Control Dyn* 24(4):710–715
- Jardin MR, Bryson AE (2012) Methods for computing minimum-time paths in strong winds. *J Guidance Control Dyn* 35(1):165–171
- Klöckner A, Pinto N, Lee Y, Catanzaro B, Ivanov P, Fasih A (2012) PyCUDA and PyOpenCL: a scripting-based approach to GPU run-time code generation. *Parallel Comput* 38(3):157–174
- Marchidan A, Bakolas E (2016) Numerical techniques for minimum-time routing on sphere with realistic winds. *J Guidance Control Dyn* 39(1):188–193
- NVIDIA Corporation (2010) CUDA Programming Guide
- Soler M, Olivares A, Staffetti E (2015) Multiphase optimal control framework for commercial aircraft four-dimensional flight-planning problems. *J Aircraft* 52(1):274–286
- Sridhar B, Ng H, Chen N (2011) Aircraft trajectory optimization and contrails avoidance in the presence of winds. *J Guidance Control Dyn* 34(5):1577–1584
- Steiner M, Bateman R, Megenhardt D, Liu Y, Xu M, Pocerlich M, Krozel J (2010) Translation of ensemble weather forecasts into probabilistic air traffic capacity impact. *Air Traffic Control Q* 18(3):229–254

Multi-objective Optimization of A-Class Catamaran Foils Adopting a Geometric Parameterization Based on RBF Mesh Morphing



Marco Evangelos Biancolini, Ubaldo Cella, Alberto Clarich
and Francesco Franchini

Abstract The design of sailing boats appendages requires taking in consideration a large amount of design variables and diverse sailing conditions. The operative conditions of dagger boards depend on the equilibrium of the forces and moments acting on the system. This equilibrium has to be considered when designing modern fast foiling catamarans, where the appendages accomplish both the tasks of lifting up the boat and to make possible the upwind sailing by balancing the sail side force. In this scenario, the foil performing in all conditions has to be defined as a trade-off among contrasting needs. The multi-objective optimization, combined with experienced aerodynamic design, is the most efficient strategy to face these design challenges. The development of an optimization environment has been considered in this work to design the foils for an *A-Class* catamaran. This study, in particular, focuses on the geometric parameterization strategy combined with a mesh morphing method based on Radial Basis Functions, and managed through the workflow integration within the optimization environment.

M. E. Biancolini · U. Cella
University of Rome “Tor Vergata”, Rome, Italy

U. Cella (✉)
Design Methods, Messina, Italy
e-mail: ubaldo.cella@designmethods.aero
URL: <https://www.designmethods.aero>

A. Clarich
ESTECO, Trieste, Italy
e-mail:
URL: <https://www.esteco.com>

F. Franchini
EnginSoft, Florence, Italy
e-mail:
URL: <https://www.enginsoft.com>

Introduction

“*Foiling*” (term used to describe a sailing condition in which the boat is lifted up from the water by lifting surfaces) is not a new idea in the sailing world [the first known sailing hydrofoil was produced in 1938 by Robert Rowe Gilruth and Carl William Price (Sheahan 2013)] but, as it often occurred for many innovative solutions, the efficient exploitation of its potentialities was related to the technological improvements in materials, manufacturing processes, design capability, etc. The foiling solutions adopted in the last America’s Cup class catamarans (called AC72) gave a strong impulse to the evolution of smaller multihull classes. The A-Class catamaran has benefited from these experiences and have shown significant innovations in the last few years due to its large diffusion and open rules.

The A-Class, born in the late 50s, is a small high-tech catamaran that is considered the fastest single-handed racing dingy in the world. The rules are very simple. They mainly constrain the minimum weight (75 kg), the hulls length (18 ft) and the sail surface (150 ft²). In 2009 new rules were added with the intention of preventing foiling for A-Cats. Given that the concept of *hydrofoiling prohibition* was ambiguous and difficult to define, an indirect approach was chosen. The idea was to introduce a set of constraints aimed to limit the surfaces suitable for sustaining the boat so to return unfavourable a flying configuration compared to a traditional one. The constraints were defined assuming a reference vertical force coefficient that, in absence of previous experience, was evaluated from the operative conditions of the Moth class foils (the estimated lift coefficient was 0.4). Such assumption showed to be conservative and allowed the development of very favourable flying configurations. The rules, therefore, make the foil dimensioning a strongly constrained design problem for which efficient implementation of multi-objective optimizations might represent the key strategy to design configurations able to broaden the range of sailing conditions in which flying boats are faster than conventional ones.

Novel solutions were traditionally tested, in A-Cats, with empirical *trial and error* approach. The improvement and the availability of engineering numerical tools (CFD, FEM, MDO...), combined to the increase of the computational resources power, contribute today to reduce the costs of advanced engineering methodologies (which in the recent past were limited to research contests or to high technological fields as the aerospace). Highly specialized engineering services are then now beginning to be compatible with the requirements of relatively limited markets as the sports dinghies. For this reason, a joint project including the university of Rome “*Tor Vergata*”, the aerospace engineering consulting firm *Design Methods* and the software vendors *RBF Morph*¹ and *ESTECO*, has been developed to setup a pilot study to demonstrate capabilities and the potentialities of combining cutting edge mesh morphing technologies and optimization design environments by developing a highly constrained multi-objective optimization procedure.

The implementation of strongly constrained geometric parameterization often suggests adopting a parametric CAD system coupled to a numerical domain regen-

¹www.rbf-morph.com.

eration procedure. In this paper, we want to demonstrate the efficiency of the mesh morphing approach based on Radial Basis Functions (using *RBF Morph*).

The optimization procedure consists in combining two-phases CFD simulations of the foils, using the *ANSYS Fluent* solver, with the mesh morphing tool *RBF Morph* within the *ESTECO modeFRONTIER* optimization workflow. The design variables control the foil planform and the front shape subjected to geometrical constraints. The objective functions are defined to improve the performances in upwind (navigation against the wind) and downwind (navigation with the wind) sailing conditions at two values of boat speed.

Shape Parameterization by Mesh Morphing

The geometric parametrization based on mesh morphing consists in implementing shape modifiers, amplified by parameters that constitute the problem variables, directly on the computational domain. New geometric configurations are obtained imposing the displacement of a set of mesh regions (e.g. walls, boundaries or discrete points within the volume) by using algorithms able to smoothly propagate the prescribed displacement to the surrounding volume. The performances of the morphing action (in terms of quality of the morphed mesh and computational resources requirements) depend on the algorithm adopted to perform the smoothing of the grid. Among the several algorithms available in literature, Radial Basis Functions are recognized to be one of the best mathematical framework to deal with the mesh morphing problem (Jakobsson and Amoignon 2007).

The first commercial mesh morphing software based on Radial Basis Functions was *RBF Morph*. The tool was born as an add-on of the *ANSYS Fluent* CFD code, fully integrated in the solving process, and was launched to the market in 2009 (Biancolini et al. 2009). Its efficiency was successfully demonstrated on several industrial engineering problems (shape optimization, ice accretion, static and dynamic FSI analyses) (Cella et al. 2016) including application with sails (Viola et al. 2015) and structural problems (Biancolini 2014). Today *RBF Morph* is also suitable as a stand-alone tool to be coupled with any solver.

Several advantages are related to the RBF mesh morphing approach:

- there is no need to regenerate the grid;
- the robustness of the procedure is preserved;
- its meshless nature allows to support any kind of mesh typology;
- the smoothing process can be highly parallelizable;
- the morphing action can be integrated in any solver.

The latter feature offers the very valuable capability to update the computational domain “on the fly” during the progress of the computation.

The main disadvantages of RBF mesh morphing methods are the requirement of a “back to CAD” procedure, some limitation in the model displacement amplitude, due to the distortion occurring after extreme morphing, and the high computational

cost related to the solution of the RBF system that, if large computational domains are involved, imposes the implementation on HPC environments.

Radial Basis Functions

Radial Basis Functions (RBF) are powerful mathematical functions able to interpolate, giving the exact values in the original points, functions defined at discrete points only (source points). The interpolation quality and its behaviour depends on the chosen RBFs. A linear system (of order equal to the number of source points introduced) needs to be solved for coefficients calculation. Once the unknown coefficients are calculated, the motion of an arbitrary point inside or outside the domain is expressed as the summation of the radial contribution of each source point (if the point falls inside the influence domain). An interpolation function composed by a radial basis and a polynomial is defined as follows:

$$s(x) = \sum_{i=1}^N \gamma_i \phi(\|x - x_i\|) + h(x) \quad (1)$$

The minimal degree of polynomial h depends on the choice of the basis function. A unique interpolant exists if the basis function is a conditionally positive definite function. If the basis functions are conditionally positive definite of order $m = 2$, a linear polynomial can be used:

$$h(x) = \eta + \eta_1 x + \eta_2 y + \eta_3 z \quad (2)$$

The values for the coefficients γ of RBF and the coefficients η of the linear polynomial can be obtained by solving the system

$$\begin{pmatrix} M & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \gamma \\ \eta \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix} \quad (3)$$

where g are the known values at the source points. M is the interpolation matrix defined calculating all the radial interactions between source points

$$M_{ij} = \phi(\|x_{k_i} - x_{k_j}\|) \quad 1 \leq i \quad j \leq N \quad (4)$$

and P is the constraint matrix

$$P = \begin{pmatrix} 1 & x_{k_1}^0 & y_{k_1}^0 & z_{k_1}^0 \\ 1 & x_{k_2}^0 & y_{k_2}^0 & z_{k_2}^0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_N}^0 & y_{k_N}^0 & z_{k_N}^0 \end{pmatrix} \quad (5)$$

The radial basis is a meshless method suitable for parallel implementation. In fact, once the solution is known and shared in the memory of each node of the cluster, each partition has the ability to smooth its nodes without taking care of what happens outside because the smoother is a global point function and the continuity at interfaces is implicitly guaranteed.

RBF Morph Setup

RBF Morph allows to extract and control points from surfaces and edges, to put points on primitive shapes (boxes, spheres and cylinders) or to specify them directly by individual coordinates and displacements. Primitive shapes can be combined in a Boolean fashion limiting the action of the morpher itself. The shape information coming from an individual RBF setup are generated interactively with the help of the GUI and are used subsequently in batch commands that allow to combine many shape modifications in a non-linear fashion (non-linearity occurs when rotation axis are present in the RBF setup). The displacement of the prescribed set of source points can be amplified according to parameters that constitutes the parametric space of the model shape.

The definition and the execution of a morphing action is completed by the following steps: *setup*, *fitting* and *smoothing*. The setup consists in the manual definition, from the program GUI, of the domain boundaries within which the morphing action is limited to, in the selection of the source points where fixed and moving mesh regions are imposed, and in the definition of the required movements of the points used to drive the shape deformation. During the fitting process, the RBF system, derived from the problem setup, is solved and stored into a file ready to be amplified. This operation has to be performed only once for every RBF problem. Stored RBF solutions are very light (in terms of files dimension) compared to storing all the created morphed mesh. The smoothing action (surfaces and volumes morphing according to arbitrary amplification factors) is first performed applying the prescribed displacement to the grid surfaces and then smoothly propagating the deformation to the surrounding domain volume. It can be performed combining several RBF solutions, each one defined by a proper amplification factor, to constitute the parametric configuration of the computational domain. Figure 1 reports an example (in this case applied to the sail) of and RBF problem setup.

Fig. 1 Fixed (red) and moving (green) source points of an RBF setup

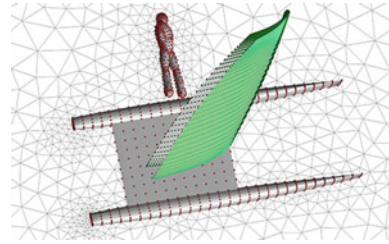
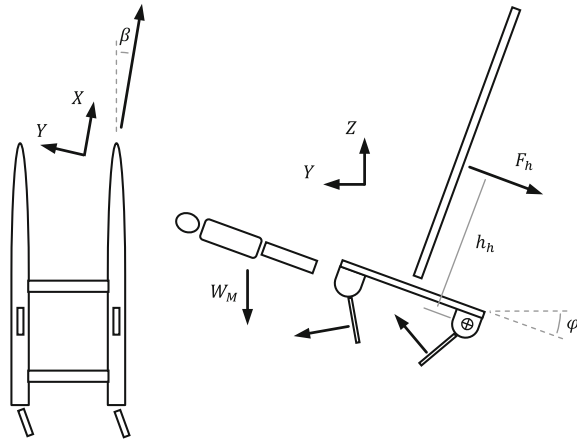


Fig. 2 Forces acting on the boat



Foils Design Problem Description

The operative conditions of sailing boats appendages depend on the equilibrium of the forces and moments acting on the system (Larsson and Eliasson 1997). The speed is related to the performances of sails and to the characteristics of the boat with a complex interaction whose estimation require to model the several aspects of the physics involved. For this reason the design of any components should be, in general, approached within so-called VPP (Velocity Prediction Program) environments (Claughton et al. 1998) (Fig. 2).

To define the design conditions of the A-Cat foils, some simplifications has been, however, adopted in this work. The vertical force equilibrium is mainly dominated by the weight of the boat and the crew. The modulus of the other components, derived from the 6DoF equilibrium, varies in a range that is, in general, smaller than the range of possible crew weight. It is then considered acceptable for the foils to assume a fixed target vertical component of the lift. Similar assumptions are accepted for the side force since it is mainly limited by the maximum righting moment generated by the helmsman at the trapeze (for a fixed known height of the sail centre of effort h_h). The task is to identify the shape of the foils that, while respecting the imposed constraints and generating the required lifting force components, minimize the drag (Strologo 2015).

Geometric Constraints Definition

A-Class rules state that all foils must be inserted from the top of the hull (to prevent the adoption of *T-foils*) and that the minimum distance between the tips must always be larger than 1.5 m (to limit the span of surfaces contributing to the vertical lift). The maximum beam of the boat, including appendages in all positions, must be lower than 2.3 m. In order to insert the foils, furthermore, a minimum value of the angle δ , assuming *L-shaped* foils, is required (Fig. 3). Finally, structural requirements impose a minimum value of the foil thickness.

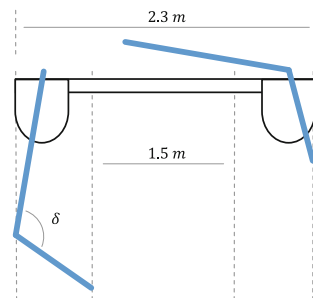
Setup of Numerical Model

A two-objective optimization procedure was setup within the *modeFRONTIER* software environment. The objectives are defined by the minimization of the boat hydrodynamic drag, excluding the rudders, in upwind and downwind sailing conditions. In downwind sailing the boat is expected to be fully lifted up from the water by the foils. Configurations that are not able to generate sufficient lift are rejected. In upwind sailing, the boat is expected to be only partially sustained by the foils.

CFD Configuration

A steady incompressible analysis, using a volume of fluid (VOF) technique to model the two-phases (air and water), was setup for the downwind analysis. The boat was assumed to sail at a heeling angle (angle φ of Fig. 2) of five degree and at a speed of 15 knots. The sinkage was iteratively trimmed to define the attitude that generates

Fig. 3 Scheme of foils constraints



the target vertical force. No cavitation model was activated.² The total displacement was assumed equal to 170 kg (empty boat weight plus crew). Considering around 30% of this value to be generated by the *T-foils* of the two rudders, the main foils were then assumed to contribute with the generation of 120 kg to the sustainment of the boat. The operative leeway angle (angle β of Fig. 2) should be defined from the global equilibrium of forces and moments acting on the boat. In was, however, here considered acceptable to keep it fixed to 3°. The proper estimation of its value would have, in fact, significantly increased the computational burden since it requires to introduce an additional degree of freedom. The balance between the additional computational cost required and the impact this simplification is expected to have on the optimization trend fully justifies, in our view, this choice.

The analysis in upwind sailing was performed at a speed of 10 knots and at a fixed attitude maintaining the computational domain unchanged (also in this case it was assumed five degrees as heeling angle). One hull is flying while the other one is floating and contributing to the sustainment. A single phase CFD analysis was setup assuming the top inviscid wall boundary of the domain (which, in order to partially account for hull/foil junction interference effects, includes a shape similar to the immersed hull) to represent the water free surface considered as planar (Fig. 4). This simplification forces to neglect effects as ventilation or hull boundary layer interference introducing uncertainties on the solution. It is, however, considered acceptable, for the optimization purpose, since the aim to estimate the drag difference between candidate solutions is prevalent on the necessity of an accurate definition of the absolute value of drag. The missing drag component of the hull is recovered by an analytical formulation developed by a comparison with a matrix of CFD solutions obtained on the isolated demihull at several attitudes and displacements [a description of the formulation adopted is reported in (Cella et al. 2016)]. The lift fraction obtained subtracting the lift generated by the foils from the boat operative displacement is used to feed the hull analytical drag model, whose output is added to the foils drag fraction to generate the objective function.

The accurate evaluation of the leeway angle is considered to be important in upwind sailing and adjusted by changing the inflow direction on the far field boundaries. Its operative value is estimated performing two preliminary analyses at two angles and then linearly extrapolating the final leeway angle at which the candidate geometry generates the required target side force. If the target side force is not generated at the expected angle, the selected configuration is rejected because it does not perform in the linear region of the aerodynamic lift polar. The target side force (in our case defined equal to 70 kg) was estimated from the equilibrium of moments around the sailing direction assuming a value for the height of the sailing centre of effort (distance h_h of Fig. 2) of 4 m.

Computational domain

A multi-block structured hexahedral mesh was generated modelling a domain extended up to ten meters upstream and downstream the foils. It is ten meters wide

²The cavitation critical C_p , at the selected downwind speed, is around -3 (Hoerner 1965). Such value is not expected to be reached in the design conditions (especially if laminar airfoils are adopted).

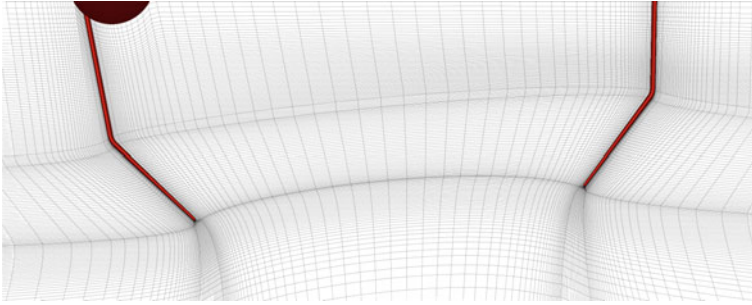


Fig. 4 Detail of the computational domain (medium mesh)

Fig. 5 Surface cells clustering for the three levels of grid

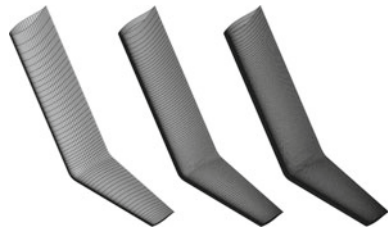
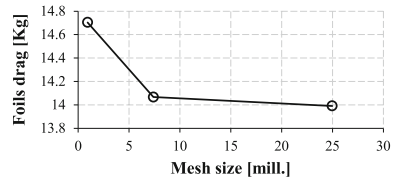


Fig. 6 Solution sensitivity to the grid dimension



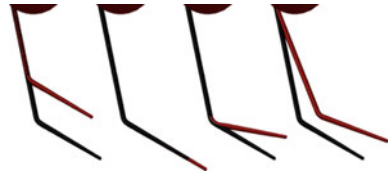
and five meters deep. The top of the domain coincides with the water level in upwind conditions. Three levels of grid were generated (Fig. 5) with the aim to evaluate the sensitivity of the solution on the grid dimension. The size of coarse, medium and fine meshes were approximately 1, 7.5 and 25 million of cells.

Figure 6 reports the solutions obtained, on the baseline geometry, with the three meshes in downwind configuration (VOF analysis trimming the sinkage to maintain the vertical lift component unchanged). The difference between the drag obtained with the coarse grid and the drag obtained adopting the fine mesh is in the order of 5% while the adoption the medium grid led to a difference limited to half percent. The coarse mesh was the one used in the optimization procedure.

Implementation of Shape Parameterization

The reference geometry is made by two straight segments smoothly blended in the junction region. The connection with the hulls is located at the external side and both

Fig. 7 Foils front shape modifiers



inner and outer segments are oriented inboard. This configuration is assumed to offer flying stability advantages (which are not accounted by the optimization criterion) with respect to an inner segment bent outboard (as the sketch in Fig. 3) (Caughey 2011). The location of the hull/foils junction is then not a design variable.

The foils segments are generated by a straight untwisted extrusion of the well-known NACA 63-412 laminar airfoil.³ The inner section is assumed to have a constant chord while the outer is tapered.

The variables of design were:

1. total foil draft;
2. outer segment cant angle (angle δ of Fig. 3);
3. angle of the inner segment respect to vertical;
4. inner segment chord (keeping the absolute foil thickness unchanged);
5. outer segment taper ratio;
6. foils sweep angle.

The last parameter is not exactly a shape parameter. It is a trim that has a direct effect on the horizontal angle of incidence of the foils. Its morphing action is implemented as a rotation of the foils along an axis perpendicular to the boat symmetry plane and passing near the hull/foil junction.

Seven shape modifiers have been setup: four to control lengths and angles of the foil segments (Fig. 7), one to set the chord of the inner segment, one for the taper ratio of the outer segment and one to control the foils sweep angle. The amplification factors of the RBF solutions are defined combining the design input variables in order to fulfil the constraints imposed by the class rules (e.g. when the cant angle δ is modified, the outer segment is scaled according to an amplification factor that recover the limits reported in Fig. 3). The morphing actions are applied in sequence and limited to a volume surrounding the foils region.

Integration in the Optimization Environment

The multi-objective design software *modeFRONTIER* allows to integrate different computational codes (any commercial or in-house tools) into a common design envi-

³The airfoil design is not included in this phase. The foils are analyzed by fully turbulent RANS analyses with the view to demand the verification of the laminar stability and the proper operating range of the airfoil to a following design stage.

Fig. 8 Pareto solution of the final two-objectives optimization

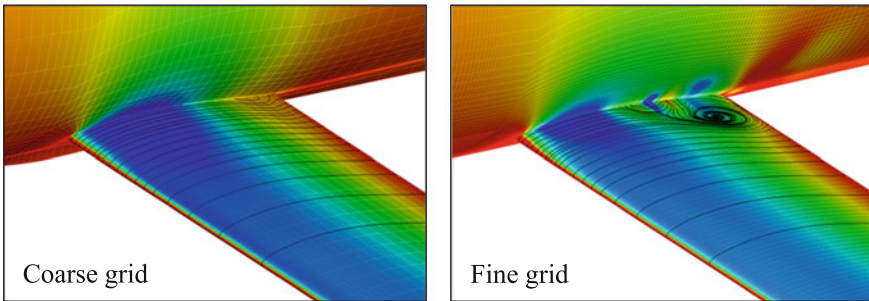
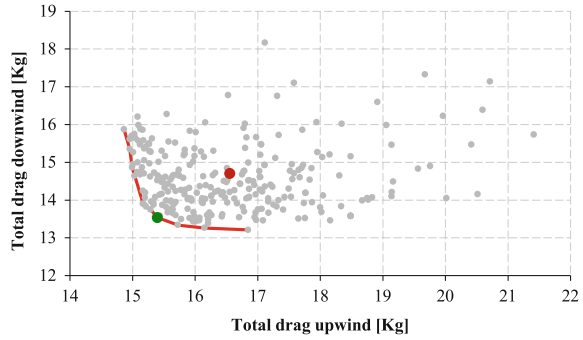


Fig. 9 Flow separation in the hull/foil junction

ronment. It allows the automatic execution of a series of designs proposed by a selected optimization algorithm (including Evolutionary Algorithms, Game Strategies, Gradient-based Methodologies, Response Surfaces, Adaptive and Automatic methodologies), up to the specified objectives are satisfied. In this modular environment, each component of the optimization process, including input variables, input files, scripts or direct interfaces to run the software, output files, output variables and objectives, is defined as a node to be connected with the other components (Vernengo 2014). The complete logic flow from parameterization to performance evaluation is defined by the user who can select among several available optimization algorithms, according to the defined objectives. Statistical and visualization tools, can then be used for an efficient decision making, allowing the designer to select the optimal configuration of the system (Clarich et al. 2013; Bonci et al. 2015).

The workflow implemented for the optimization of the A-Class foils followed the scheme reported in Fig. 10 in appendix. The starting reference geometry is updated each cycle, by the morphing procedure described above, according to the design variables selected by the decision making criterion. The candidate evaluation process is managed by a script procedure written in *Scheme* language. The analyses at the two sailing conditions are performed in sequence. The upwind analysis is run if the downwind analysis is successful.

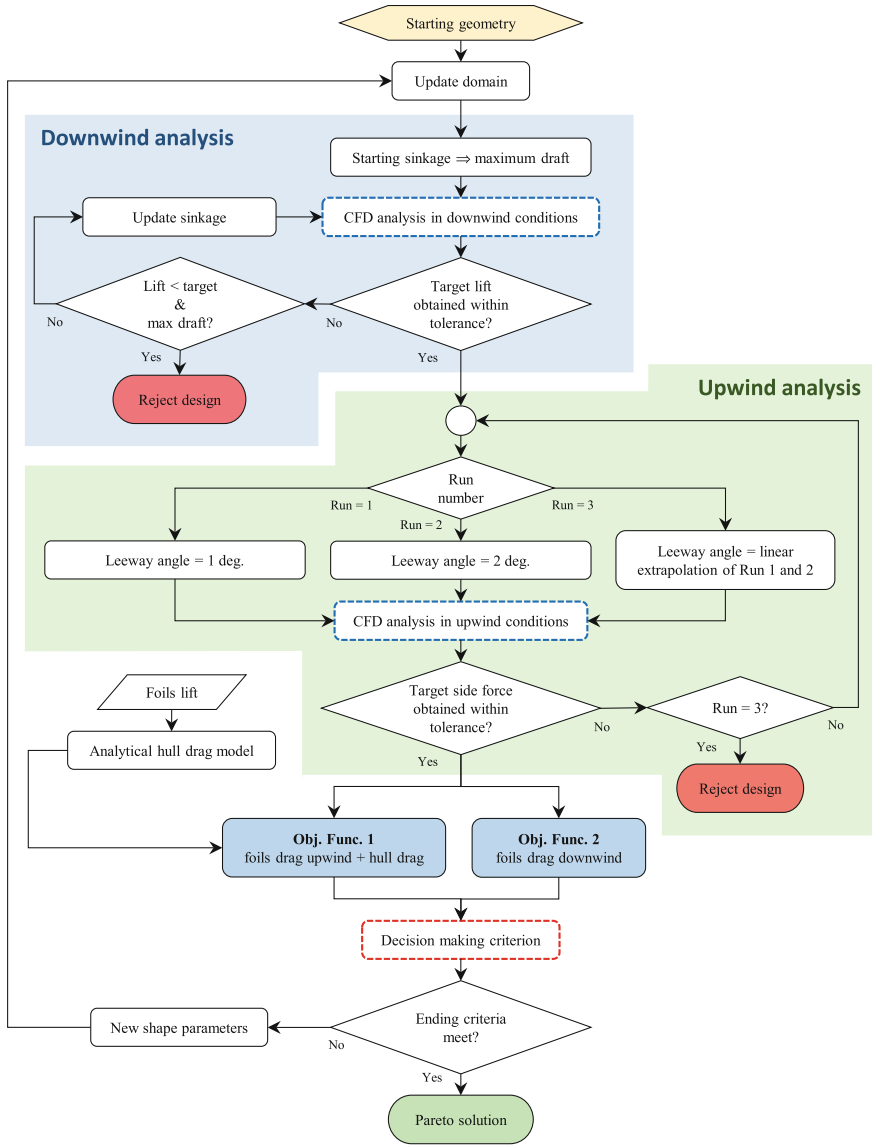


Fig. 10 Free surface in downwind sailing by baseline (left) and optimum (right)

The downwind analysis begins at the maximum sinkage (hull flying around 15 cm from the water surface). If the lift generated is higher than target, the computation progresses trimming the sinkage up to the vertical equilibrium, otherwise the design is rejected. In upwind conditions, as stated, three computations are run to select the

leeway angle that generates the required side force. If the final solution do not lay in the linear aerodynamic polar region, the candidate is rejected.

A two-objectives optimization was performed adopting the MOGA-II, a proprietary version of the Multi-Objective Genetic Algorithm (Quagliarella et al. 1996). The two defined objective functions were the minimization of the total drag at the two sailing conditions. The evaluation of the hull drag fraction in upwind condition was included in the *modeFRONTIER* environment by a node that, after the foils CFD analyses, executes the analytical hull drag model developed in form of a *Scilab* function.

Solutions

The time elapsed to complete the evaluation of one valid design, using the coarse mesh, ranged between 15 and 20 min on a workstation equipped with 20 CPU (2 processors Intel Xeon E5-2680 2.8 GHz with 10 cores each). The time required for the morphing action was less than 2 min. More than 400 evaluations were performed in three days. Among them about 40% of design candidates were rejected because of failure in the minimum lift requirement criterion. The solution obtained is reported in Fig. 8.

The green point on the Pareto front is the optimum solution which is considered the best compromising design. The red circle refers to the starting baseline geometry which was built roughly referring to existing designs. The estimated drag reduction in upwind sailing is 7% (hull plus foils) while in downwind is 7.9%.

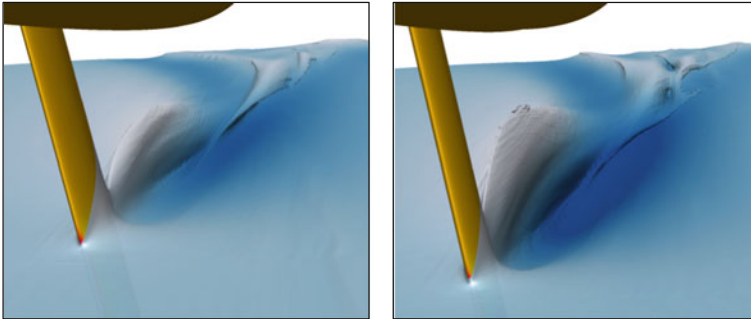
Post Design Verification

The selected optimum was verified using the fine mesh adopted for the grid sensitivity evaluation. The RBF solutions were applied to the fine baseline grid (the method is meshless) to obtain the fine mesh of the optimum geometry. The analysis also allowed to verify if the evaluation of the improvement is confirmed. The results of this verification are summarized in Table 1 for both downwind and upwind sailing conditions.

The improvement was overestimated by only 0.24% in downwind sailing. This data alone should confirm the coarse grid, despite the lower absolute accuracy it involves, to be suitable to correctly drive the optimization process toward the optimum. The improvement in upwind condition was, conversely, significantly overestimate. The total drag computed adopting the fine grid is, furthermore, slightly higher than the drag estimated adopting the coarse mesh. The reason of this behaviour has to be ascribed to a large separation observed in the hull/foil junction, on both baseline and optimized geometries, that the coarse mesh is not able to capture (Fig. 9). The direction toward the optimum, and the validity of the optimization solution, should

Table 1 Foils drag solutions in downwind conditions

	Mesh	Baseline (kg)	Optimized (kg)	Drag reduction (%)
Downwind	Coarse	14.7	13.54	7.89
	Fine	13.99	12.92	7.65
Upwind	Coarse	16.55	15.4	6.96
	Fine	16.85	16.5	2.08

**Fig. 11** Flowchart of the optimization procedure

not be significantly affected but a finer grid in the foil root region would provide a more accurate configuration. Figure 11 in appendix report the visualization of the free surfaces generated by the baseline and the optimized solutions in downwind conditions.

Conclusions

A design procedure, based on multi-objective optimization, has been presented. The core of the method is the parameterization of the geometry implemented by a mesh morphing technique based on Radial Basis Functions. A pilot study has been setup to prove the capability of the RBF parametrization approach to face complex and strongly constrained design problems. The foils of an A-Class catamaran have been optimized at two sailing conditions. A multi-objective optimization, using genetic algorithms, was setup within the *modeFRONTIER* environment. The analysis of candidates was implemented by a script procedure used to:

- drive the morphing of the numerical domain, according to the variables of design, by the *RBF Morph* tool;
- run in sequence the computations at the two sailing conditions trimming the attitudes to generate the required side force (in upwind sailing conditions) and vertical lift component (in downwind sailing);

- extract the information required to compute the objective functions.

The script is executed within the *ANSYS Fluent* CFD solver. The target of design was the minimization of drag in the two operating conditions. During upwind sailing, the boat is not supposed to fly. The total drag was, in this condition, integrated adding the drag component of the hull estimated by an analytical model (rudders are excluded) and integrated in the process by a node in *modeFRONTIER* that execute a function in the *Scilab* environment.

The optimization process led to a Pareto front on which a compromising design, that improved the performance by 7% in upwind conditions and by 7.9% in downwind, has been selected. Since the main objective of the work was to demonstrate the efficiency of the proposed approach for design, a very light mesh (less than one million of hexahedral cells) was used in the optimization workflow. A post design verification of the selected optimum, adopting a very fine mesh, confirmed the improvement in downwind sailing conditions (the difference in the estimation of improvement was limited to 0.24%). In upwind sailing, a large separation in the hull/foil junction region, affected the solution verification. The coarse mesh was not able to capture the phenomenon leading to significantly resize the estimated improvement but confirming the numerical configuration validity in indicating the optimum direction.

The work demonstrated the RBF mesh morphing approach to be a very good option to face complex constrained parametrization problems. It does not require to define a parametric CAD model and offers several advantages: no re-mesh required, high robustness, high parallelizability, meshless properties. The possibility to combine several RBF solutions and to define each amplification factor according to any formulation able to account for external constraints offer large flexibility in setting up complex parameterizations. The high parallelizable feature, furthermore, extend the potentialities of the method by providing the possibility, within HPC environments, to setup optimization configurations that involve large computational domains.

References

- Biancolini ME (2014) RBF morph mesh morphing ACT extension for ANSYS mechanical. In: Automotive simulation world congress, Tokyo, October 2014
- Biancolini ME et al Industrial application of the meshless morpher RBF morph to a motorbike windshield optimisation. In: 4th European automotive simulation conference, Munich, Germany, 6–7 July 2009
- Bonci M, Viviani, M, Broglia R, Dubbioso G (2015) Method for estimating parameters of practical ship manoeuvring models based on the combination of RANSE computations and system identification. In: Applied ocean research, vol. 52. August 2015, pp 274–294
- Caughey DA (2011) Introduction to aircraft stability and control. Course Notes for M&AE 5070, Cornell University, New York, pp 14853–7501
- Cella U, Groth C, Biancolini ME (2016) Geometric parameterization strategies for shape optimization using RBF mesh morphing. In: Advances on mechanics, design engineering and manufacturing, pp 537–545, September 2016

- Cella U, Salvatore F, Ponzini R (2016) Coupled sail and appendage design method for multihulls based on numerical optimisation. PRACE—EU SHAPE Project Final Report, 5th July 2016
- Cloughton AR, Sheno R, Wellicome JF (1998) *Sailing yacht design: theory*. Addison Wesley, Longman
- Clarich A et al Ottimizzazione della regolazione di una vela rigida per catamarani da regata mediante modeFRONTIER e ANSYS. In: ANSYS User Group Meeting, Italy, June 2013
- Hoerner SF (1965) *Fluid-dynamic drag*, Hoerner fluid dynamics. Hoerner, Bakersfield, CA (US)
- Jakobsson S, Amoignon O (2007) Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Comput Fluids* 36(6):1119–1136
- Larsson L, Eliasson RE (1997) *Principle of sailing yacht design*. Adlard Coles Nautical, London, UK
- Sheahan M (2013) *High-speed sailing*. Ingenia, Royal Academy of Engineering, Issue 57
- Stroligo M (2015) Preliminary design investigation for the development of new hull shapes for America's cup class catamaran AC-62. In: International CAE conference 2015, Verona, Italy, 19–20 October 2015
- Vernengo G (2014) Design by optimization of ship hull forms. New perspectives through full parametric modelling and multi-objective optimization. In: modeFRONTIER International Users Meeting, 2014, Italy
- Viola IM, Biancolini ME, Sacher M, Cella U (2015) A CFD-based wing sail optimization method coupled to a VPP. In: 5th high performance yacht design international conference, 8–12 March 2015, Auckland (NZ)
- Quagliarella D, Périaux J, Poloni C, Winter G (1996) *Genetic algorithms and evolution strategy in engineering and computer science: recent advances and industrial applications*, Chap. 13, pp 267–288. John Wiley & Sons, Chichester, UK

Development of an Efficient Multifidelity Non-intrusive Uncertainty Quantification Method



Saeed Salehi, Mehrdad Raisee, Michel J. Cervantes and Ahmad Nourbakhsh

Abstract Most engineering problems contain a large number of input random variables, and thus their polynomial chaos expansion (PCE) suffers from the curse of dimensionality. This issue can be tackled if the polynomial chaos representation is sparse. In the present paper a novel methodology is presented based on combination of ℓ_1 -minimization and multifidelity methods. The proposed method employ the ℓ_1 -minimization method to recover important coefficients of PCE using low-fidelity computations. The developed method is applied on a stochastic CFD problem and the results are presented. The transonic RAE2822 airfoil with combined operational and geometrical uncertainties is considered as a test case to examine the performance of the proposed methodology. It is shown that the new method can reproduce accurate results with much lower computational cost than the classical full Polynomial Chaos (PC), and ℓ_1 -minimization methods. It is observed that the present method is almost 15–20 times faster than the full PC method and 3–4 times faster than the classical ℓ_1 -minimization method.

S. Salehi (✉) · M. Raisee · A. Nourbakhsh
Hydraulic Machinery Research Institute, School of Mechanical Engineering,
College of Engineering, University of Tehran, P.O. Box 11155-4563,
Tehran, Iran
e-mail: s.salehi@ut.ac.ir

M. Raisee
e-mail: mraisee@ut.ac.ir

A. Nourbakhsh
e-mail: anour@ut.ac.ir

M. J. Cervantes
Division of Fluid and Experimental Mechanics, Luleå University
of Technology, P.O. Box 97187, Luleå, Sweden
e-mail: michel.cervantes@ltu.se

Introduction

Almost all of engineering applications contain different uncertainties such as, uncertainties in physical properties, randomness in boundary and operating conditions and geometrical uncertainties due to the manufacturing tolerances. In recent years, The uncertainty quantification (UQ) techniques have received a lot of attentions due to the fast growing of computational resources. In the literature, various methods have been proposed for UQ. The Monte Carlo (MC) approach (Fishman 1996) is the oldest method used for UQ. More recently the polynomial chaos (PC) approach has been developed and widely used due to its efficiency. The method is based on spectral representation of the stochastic system output as a linear combination of orthonormal multivariate polynomials. Ghanem and Spanos (1938) have developed this method for Hermite polynomials based on homogeneous chaos theory of Wiener Wiener (1938). In recent years, Several attempts have been made to develop efficient methods to reduce the size of stochastic problem, namely reduced basis methods (Nair and Keane 2002), adaptive methods (Lucor and Karniadakis 2004; Wan and Karniadakis 2005), sparse methods (Doostan and Owhadi 2011) and multifidelity methods (Ng and Eldred 2012a). All of these efforts have been made to tackle the curse of dimensionality and reduce the computational cost of UQ. They all showed that the developed method can reproduce a comparable accuracy to the classical PC method with a lower cost.

Recently, the ℓ_1 -minimization technique has been shown to be very efficient to reduce the size of stochastic space when the solution (PCE coefficients) is sparse (Doostan and Owhadi 2011).

Another approach to reduce the computational cost of uncertainty quantification is using the multifidelity methods. The multifidelity approaches aim at managing the trade-off between fidelity and expense on computations and try to achieve accurate statistics using combination of “low-fidelity” and “high-fidelity” computations. In order to obtain the low-fidelity model evaluations, one can simplify the physics, use a coarser discretization or employ reduced-order models. Such methods have been initially developed to reduce the cost of optimization processes. Ng and Eldred (2012a) extended the multifidelity approach for non-intrusive polynomial chaos and stochastic collocation using sparse grids.

To address this issue, in the present paper a novel methodology is presented based on combination of ℓ_1 -minimization and multifidelity methods. The proposed method employ the ℓ_1 -minimization method to recover important coefficients of PCE using low-fidelity computations. If the solution is sparse the number of remained PC coefficients will be substantially lower than the full PC expansion. After that the multifidelity method is employed to correct a subset of recovered coefficients. The multifidelity ℓ_1 -minimization method is then successfully applied on two different test cases and it was shown that the the proposed method converges more rapidly than the classical ℓ_1 -minimization method.

Polynomial Chaos Expansion

Suppose $y = \mathcal{U}(\boldsymbol{\xi})$ is a physical or mathematical model, where $\boldsymbol{\xi} = \{\xi_1, \xi_2, \dots, \xi_d\} \in \mathbb{R}^d$ is the set of input variables and y is the model response or the quantity of interest (QoI). If the input vector $\boldsymbol{\xi}$ is uncertain, with joint probability density function $f(\boldsymbol{\xi})$, then model response y is also stochastic. Using homogeneous chaos theory (Wiener 1938), it has been shown in Soize and Ghanem (2004) that if input stochastic variables $\boldsymbol{\xi}$ are independent, the model response can be represented as a series of orthogonal polynomial basis. The polynomial chaos representation of the random field of order p for d random variables $\boldsymbol{\xi} \equiv \{\xi_i\}_{i=1}^d$ can be written as:

$$u(\mathbf{x}; \boldsymbol{\xi}) = \sum_{i=0}^P u_i(\mathbf{x}) \boldsymbol{\psi}_i(\boldsymbol{\xi}), \quad (1)$$

where the number of terms in the summation is:

$$P + 1 = \binom{p+d}{d} = \frac{(p+d)!}{p!d!}. \quad (2)$$

The u_i 's are unknown coefficients and the $\boldsymbol{\psi}_i(\boldsymbol{\xi})$'s are multivariate polynomials orthogonal with respect to joint probability density function $f(\boldsymbol{\xi})$:

$$\langle \boldsymbol{\psi}_i(\boldsymbol{\xi}), \boldsymbol{\psi}_j(\boldsymbol{\xi}) \rangle = \langle \boldsymbol{\psi}_i(\boldsymbol{\xi}), \boldsymbol{\psi}_i(\boldsymbol{\xi}) \rangle \delta_{ij}, \quad (3)$$

where $\langle \cdot, \cdot \rangle$ represents the inner product in Hilbert space. The input random variables $\boldsymbol{\xi} \equiv \{\xi_i\}_{i=1}^d$ are assumed to be independent, so that the probability density function can be expressed as:

$$f(\boldsymbol{\xi}) = \prod_{i=1}^d f_i(\xi_i), \quad (4)$$

where $f_i(\xi_i)$ is the PDF of ξ_i . Each basis function $\boldsymbol{\psi}_i(\boldsymbol{\xi})$ is constructed using tensor product of univariate polynomials. In order to achieve exponential convergence, the basis of the PCE should be a set of polynomials that are orthogonal with respect to the PDF of the uncertain input parameters.

Calculation of PCE Coefficients

Because the orthogonal basis is chosen, if the coefficients (u_α 's) are known, so the polynomial chaos expansion of the response is known. Thus, the problem is actually finding the coefficients. In the current study, the regression method has been used to compute the response expansion coefficients. The regression-based NIPC method starts with Eq. (1). First, one should draw a set of space filling N

realizations of input stochastic vector $\Xi = \{\xi^{(1)}, \xi^{(2)}, \dots, \xi^{(N)}\}$ and evaluate the quantity of interest for each realization $\mathcal{Y} = \{y^{(1)}, y^{(2)}, \dots, y^{(N)}\}^T$, where $y^{(i)} = \mathcal{U}(\xi^{(i)})$. Number of realizations are commonly more than the number of unknowns for numerical stability. Following Hosder et al. (2006), $N = 2(P + 1)$ vectors may be chosen in the stochastic space and the stochastic function is evaluated at these sampling points using the deterministic solver. The coefficients can then be obtained from the solution of the following over-determined linear system,

$$\Psi \mathbf{u} = \mathcal{Y}, \tag{5}$$

using the least-square approach,

$$\mathbf{u} = (\Psi^T \Psi)^{-1} \Psi^T \mathcal{Y}. \tag{6}$$

In the current study, the Sobol’ quasi-random sequence (Sobol’ 1967) is used to generate the sample points. The Sobol’ sequence is a base-2 digital sequence that fills space in a highly uniform manner. Due to the orthogonality of the basis, the mean and variance of the QoI read:

$$\mu(\mathbf{x}) = \langle \mathcal{U}(\mathbf{x}; \xi) \rangle = u_0(\mathbf{x}), \tag{7}$$

$$\sigma^2(\mathbf{x}) = \text{Var} \left(\sum_{i=0}^P u_i(\mathbf{x}) \psi_i(\xi) \right) = \sum_{i=1}^P u_i^2(\mathbf{x}) \langle \psi_i \psi_i \rangle. \tag{8}$$

Hence if the PC basis is orthonormal, the variance will be easily calculated by summing squares of PCE coefficients except the first coefficient.

ℓ_1 -Minimization for Sparse PC

If the original signal \mathbf{u} (PCE coefficients) is sparse and having obtained the measurement vector \mathcal{Y} , the sparse signal can be recovered by solving the optimization problem of the form

$$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\text{argmin}} \|\mathbf{u}\|_0 \quad \text{subject to} \quad \Psi \mathbf{u} = \mathcal{Y}, \tag{9}$$

where quasi-norm $\|\mathbf{u}\|_0$ is the number of non-zero terms in the signal \mathbf{u} .

It can be shown that the solution of the optimization problem (9) is not unique and is NP-hard to compute (Muthukrishnan 2005). Therefore, it is preferred to translate this problem into a computationally tractable problem by replacing $\|\cdot\|_0$ with its convex approximation $\|\cdot\|_1$ and minimizing the $\|\mathbf{u}\|_1$ (Davenport et al. 2011), i.e.,

$$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\text{argmin}} \|\mathbf{u}\|_1 \quad \text{subject to} \quad \Psi \mathbf{u} = \mathcal{Y}, \tag{10}$$

The p th-order polynomial chaos representation of signal \mathcal{Y} is not necessarily complete or exact (noisy signal). Therefore, a truncation error should be considered and the ℓ_1 -norm minimization of noisy signal becomes

$$\hat{\mathbf{u}} = \underset{\mathbf{u}}{\operatorname{argmin}} \|\mathbf{u}\|_1 \quad \text{subject to} \quad \|\Psi \mathbf{u} - \mathcal{Y}\|_2 \leq \varepsilon. \quad (11)$$

A large number of powerful algorithms have been proposed specifically for solving the ℓ_1 -minimization problem. All of these methods can be classified into two categories, namely Basis Pursuit (BP) and greedy algorithms. The advantages and disadvantages of these two categories are not a concern in this paper (see Needell 2009 for more information). In the present paper the greedy algorithm Orthogonal Matching Pursuit (OMP) is employed for sparse recovery.

Orthogonal Matching Pursuit

Among numerous greedy algorithms the Orthogonal Matching Pursuit (OMP) (Pati et al. 1993; Davis et al. 1997) is used in the current study due to its low computational cost and fast speed during the cross-validation procedure. If the under-determined linear system (5) is given, OMP selects the most dominant basis and then the coefficients are finally computed using the least-square approach. The OMP evaluates the relative importance of each base by projection of response vector onto the base using the inner product to find the contribution of each base to the results:

$$\frac{|\langle \psi_\alpha, \mathcal{Y} \rangle|}{\|\psi_\alpha\|_2} = \frac{1}{\|\psi_\alpha\|_2} \int_{\Omega} \mathcal{Y} \psi_\alpha(\xi) f(\xi) d\xi, \quad (12)$$

where $f(\xi)$ is the joint probability density function of Gram-Schmidt basis. Note that, the inner product in (12) is divided by norm of base to cancel the effect of base length. In each iteration of OMP the integral in Eq. (12) is evaluated numerically.

Like most of greedy algorithms, OMP consists of two important steps in each iteration: basis selection and coefficient update. OMP initializes with a residual output vector $\mathbf{r}^{(0)} = \mathcal{Y}$. In each step the most correlated base with the current residual is found. The contribution of chosen base is subtracted from output vector (or current residual) to compute a new residual. This procedure is repeated until the residual criterion is satisfied (For more details please see Salehi et al. 2017).

Multifidelity Polynomial Chaos

Assume y_{high} is the response (or QoI) of a system obtained by a expensive high-fidelity (accurate) modeling, $y_{\text{high}} = \mathcal{U}_{\text{high}}(\xi)$, and y_{low} is the response of the same

system evaluated with a inexpensive low-fidelity modeling, $y_{\text{low}} = \mathcal{U}_{\text{low}}(\boldsymbol{\xi})$. The main idea behind the multifidelity polynomial chaos is to correct the low-fidelity responses in order to match the high-fidelity model values using a correction term C (Ng and Eldred 2012b). Having obtained the polynomial chaos expansion of high and low-fidelity responses, the stochastic expansion of additive correction reads

$$C(\boldsymbol{\xi}) = \mathcal{U}_{\text{high}}(\boldsymbol{\xi}) - \mathcal{U}_{\text{low}}(\boldsymbol{\xi}), \tag{13}$$

so

$$\begin{aligned} y_{\text{high}} &\equiv \mathcal{U}_{\text{high}}(\boldsymbol{\xi}) = \mathcal{U}_{\text{low}}(\boldsymbol{\xi}) + C(\boldsymbol{\xi}) \\ &= \sum_{0 \leq |\boldsymbol{\alpha}| \leq p} u_{\alpha, \text{low}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) + \sum_{0 \leq |\boldsymbol{\alpha}| \leq p} u_{\alpha, c} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}), \end{aligned} \tag{14}$$

where $u_{\alpha, \text{low}}$ and $u_{\alpha, c}$ are PCE coefficients of low-fidelity and correction expansions respectively. The important rule is that the number of high-fidelity model evaluations should be less than low-fidelity evaluations (in most cases by order of magnitudes). Hence the indices of correction expansion must be a subset of low-fidelity expansion indices. For example, considering the total order rule for truncating the expansion, to construct a p th order multifidelity expansion, one can use a p th order low-fidelity expansion combined with q th order ($q < p$) correction expansion,

$$\mathcal{U}_{\text{multi}}(\boldsymbol{\xi}) = \sum_{0 \leq |\boldsymbol{\alpha}| \leq q} (u_{\alpha, \text{low}} + u_{\alpha, c}) \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) + \sum_{q < |\boldsymbol{\alpha}| \leq p} u_{\alpha, \text{low}} \psi_{\boldsymbol{\alpha}}(\boldsymbol{\xi}). \tag{15}$$

In this way the low-order terms of PC expansion, which usually contains most of information, are corrected while the low-fidelity computations are used only for high-order terms.

Multifidelity ℓ_1 -Minimization

As demonstrated above, the multifidelity polynomial chaos introduces an efficient UQ method where the *strong* coefficients are corrected through high-fidelity computations and *weak* coefficients are computed with low-fidelity evaluations. A disadvantage of method is that before obtaining the solution of stochastic problem, no information on the values of PCE coefficients are available. Hence, in order to use the multifidelity method one needs to make decision on which coefficients should be corrected via high-fidelity computations. As mentioned in Section “Multifidelity Polynomial Chaos”, using the total order rule for truncation, one could correct the coefficients corresponding to the low-order basis in the PC expansion assuming that the PCE coefficients always decay with increasing order. However, this assumption does not always leads to an accurate multifidelity PC expansion. For

example consider sparse PC expansions where limited number of *strong* coefficients are available in the expansion that may correspond to the low or high-order basis.

As discussed earlier, the ℓ_1 -minimization method can efficiently recover the important coefficients in the sparse polynomial chaos expansions. Here the main idea behind the multifidelity ℓ_1 -minimization method is to first obtain information about the strengths of coefficients in PC expansion using low-fidelity computations. If the solution is sparse the number of remained PCE coefficients will be substantially lower than the full PC expansion. After finding the important indices a criterion is needed to decide which coefficients need to be corrected via high-fidelity computations. The criterion can be defined based on the energy of the coefficients. It means among all coefficients correct those with higher energy than a fraction of average energy, i.e.

$$\mathcal{I}_{\text{high}} = \left\{ i : |u_{\alpha_i}| \geq \beta \frac{\sqrt{\sum_{j=1}^n u_{\alpha_j}^2}}{n} \right\} \quad (16)$$

where β is the threshold parameter specifying how many terms needs to be corrected using high-fidelity computations. The threshold parameter β can vary in the range of $0 \leq \beta \leq \beta_{\text{max}}$, where β_{max} is defined as

$$\beta_{\text{max}} = \frac{n |u_{\alpha, \text{max}}|}{\sqrt{\sum_{j=1}^n u_{\alpha_j}^2}}, \quad (17)$$

and $u_{\alpha, \text{max}}$ is norm of strongest coefficient. Decreasing β results in higher number of indices to be corrected, requiring more high-fidelity calculations and higher accuracy in the multifidelity PC expansion. In contrast, increasing β parameter decreases the subset size of high-fidelity indices and total cost of multifidelity computations. When $\beta = 0$, all of coefficients (from the strongest to the weakest one) will be corrected, while with $\beta = \beta_{\text{max}}$, only the strongest coefficient is corrected. Note that the accuracy of low-fidelity computations are important to obtain an accurate multifidelity expansion. In fact, the low fidelity computations are not supposed to yield a precise QoI, but they should capture the trends well.

Results and Discussion

The performance of developed method is examined using the stochastic transonic flow around RAE2822 airfoil. Complex flow features, such as compressibility effects, boundary layer and normal shock wave formation make this flow a challenging test case for examining the performance of UQ methods. For this reason, in the literature, many researchers have used the RAE2822 as an uncertainty quantification test case (Witteveen et al. 2009; Salehi et al. 2017). The detailed geometry and flow configuration of the RAE2822 test case is described in Cook et al. (1977).

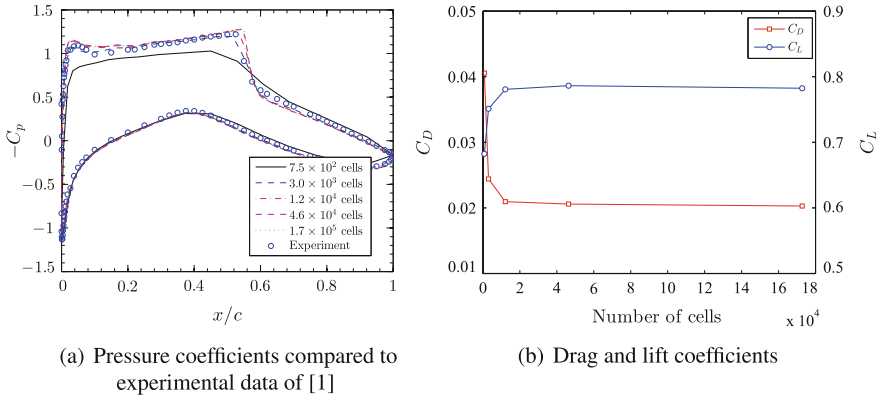


Fig. 1 Effect of mesh refinement on the numerical results

The off-design nominal flow conditions correspond to free-stream Mach number of $M = 0.734$, angle of attack of $\alpha = 2.79^\circ$, and Reynolds number of $Re = 6.5 \times 10^6$. The 2D compressible RANS equations are solved using the Spalart-Allmaras turbulence model. The second-order upwind scheme is used for discretization of the convective terms in all transport equations. First, the effect of mesh refinement is studied using five different grids (from the coarse mesh with 7.5×10^2 cells to the fine mesh with 1.7×10^5 cells). Figure 1a compares the airfoil pressure coefficients for different grids. In this figure, the numerical results are also compared with the experimental data of Cook et al. (1977). It is observed that, the 7.5×10^2 cells mesh is not able to reproduce the correct C_p distribution on the suction side of the airfoil, specially around the shock wave. On the other hand computation using finer meshes predicts almost the same trend with slight differences around sharp pressure changes across the shock region. It is noted that the predicted pressure coefficient distributions on 4.6×10^4 and 1.7×10^5 grids are identical. In addition, Fig. 1b shows mesh refinement affects drag and lift coefficients. Here it is also seen that the 4.6×10^4 cells mesh is fine enough to produce correct drag and lift coefficients. Hence the 4.6×10^4 cells mesh is chosen for the high-fidelity evaluations in the UQ analysis.

In the present study the uncertainties are imposed on operational conditions as well as airfoil geometry, to create a test case with high-dimensional stochastic space. The Mach number and angle of attack of far field flow are considered random as the operational uncertainties with means similar to the off-design nominal conditions (i.e. $M = 0.734$ and $\alpha = 2.79^\circ$) and standard deviations $\sigma_M = 0.005$ and $\sigma_\alpha = 0.1^\circ$. The geometrical uncertainties due to the manufacturing errors are modeled as a stochastic process perturbation in the surface normal direction. The stochastic process is assumed to have a zero mean and a covariance function of form:

$$R(s_1, s_2) = \sigma^2 e^{-|s_1 - s_2|/l}, \tag{18}$$

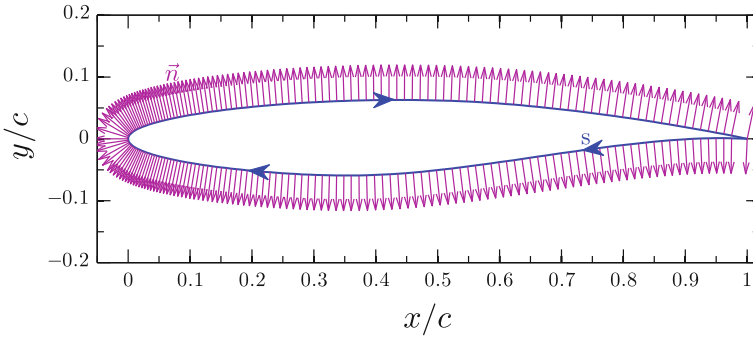


Fig. 2 s coordinate and unit normal vectors of RAE2822 airfoil

where s_1 and s_2 are positions along airfoil and l is the correlation length. The s coordinate (depicted in Fig. 2) is zero at the trailing edge, increases along the lower surface toward the leading edge and along the upper surface from leading edge toward the trailing edge. For RAE2822 airfoil $0 \leq s \leq 2.032$. Using the Karhunen-Loève (KL) expansion, the geometrical uncertainties on airfoil surface can be described as:

$$X(s; \boldsymbol{\xi}) = \bar{X}(s) + \sum_{n=1}^m \sqrt{\lambda_n} \phi_n(s) \xi_n \cdot \mathbf{n}(s), \tag{19}$$

where $\bar{X}(s)$ is the mean (nominal) airfoil coordinates at locations s , \mathbf{n} is the unit normal vector of airfoil surface, λ_n and ϕ_n are eigenvalues and eigenfunctions of covariance kernel and m is number of terms in the truncated expansion. For correlation length $l = 0.2$ and standard deviation $\sigma = 0.002$ the analytical eigenvalues and eigenfunctions are displayed in Fig. 3. In this figure, the decay rate of eigenvalues is mild due to the average correlation lengths ($l = 0.2$). Thus, many terms are needed in the truncated KL expansion. The eigenfunctions, $\phi_n(s)$, corresponding to the first five terms of KL expansion show that with the increase in index n , the modes frequency increase. Using the criterion:

$$\frac{\sum_{n=1}^m \lambda_n}{\sum_n \lambda_n} \geq 0.95, \tag{20}$$

the first 20 eigenvalues and eigenfunctions are needed in the KL expansion, which implies that 20 random variables are used to define uncertainty in the airfoil geometry.

Considering operational and geometrical uncertainties, a total of 22 random variables ($d = 22$) are present in the analysis. All random variables are set to be uniformly distributed on stochastic space $[-1, 1]^d$. The UQ analyses are carried out using a second order Legendre polynomials basis. In addition an over-sampled full PC

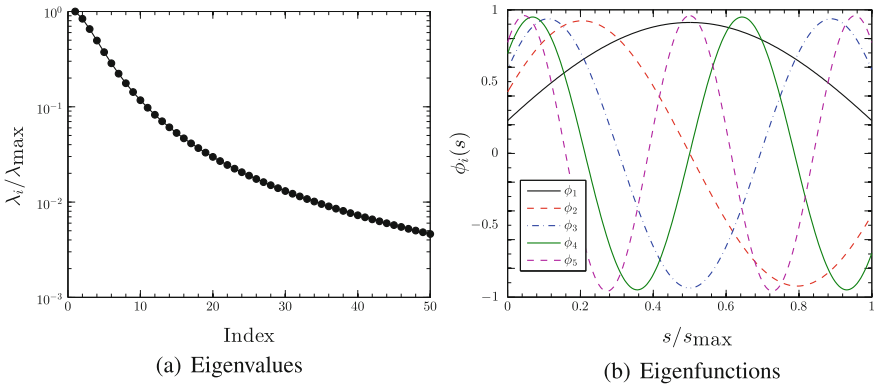


Fig. 3 Eigenvalues and eigenfunctions of covariance kernel for $l = 0.2$ and $\sigma = 0.002$

analysis with polynomials order $p = 3$ is carried out using 4600 CFD simulations as a reference solution. The errors are computed with respect to this reference solution.

At this point another mesh should be chosen for the low-fidelity analysis. It was observed that the mesh with 7.5×10^2 cells is not able to capture the exact C_p distribution. As shown in Fig. 1a, although computation on the coarsest mesh with 7.5×10^2 grid cells does not yield entirely correct C_p 's, predictions on 3.0×10^3 mesh yield acceptable C_p distribution. Also, the drag and lift coefficients of this mesh are in good agreement with the fine mesh (Fig. 1b). From another prospective, a low-order UQ analysis could help to select the low-fidelity analysis mesh. In the present study, full PC UQ analysis were performed for all grids with $p = 1$, using 46 CFD samples. Figure 4 indicates that the first ten PCE coefficients of stagnation point and normal shock pressure coefficient obtained with 7.5×10^2 cells mesh are much different with other grids and finer grids can produce similar PCE coefficients. Similar trends were also observed for the pressure coefficient at other points, as well as drag and lift coefficients. Therefore, this analysis also shows that the 3.0×10^3 cells mesh is fine enough to be used for the low-fidelity computations. It should be noted that the convergence criteria of 10^{-5} and 10^{-4} are employed for the high-fidelity and low-fidelity CFD computations respectively.

The effectiveness of developed method is assessed by comparing convergence rate to the classical ℓ_1 -minimization and full PC methods. The errors in mean and variance of drag and lift coefficients of RAE2822 airfoil calculated with different methods are compared in Fig. 5. While the full PC analysis shows a slow convergence, the ℓ_1 -minimization and multifidelity ℓ_1 -minimization methods converges rapidly. The developed multifidelity ℓ_1 -minimization has the fastest convergence rate and thus for a specific error needs smallest number of high-fidelity model evaluations. For example it can be seen that for a relative error of 0.1% for mean of C_D , the full PC needs more than 500 high-fidelity CFD samples and the ℓ_1 -minimization reaches this

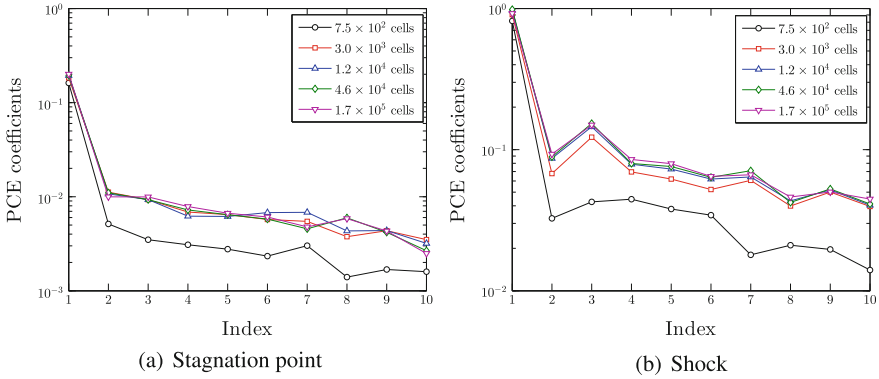


Fig. 4 First ten PCE coefficients of C_p of **a** stagnation point and **b** shock region, obtained with different grids using first order full PC UQ analysis

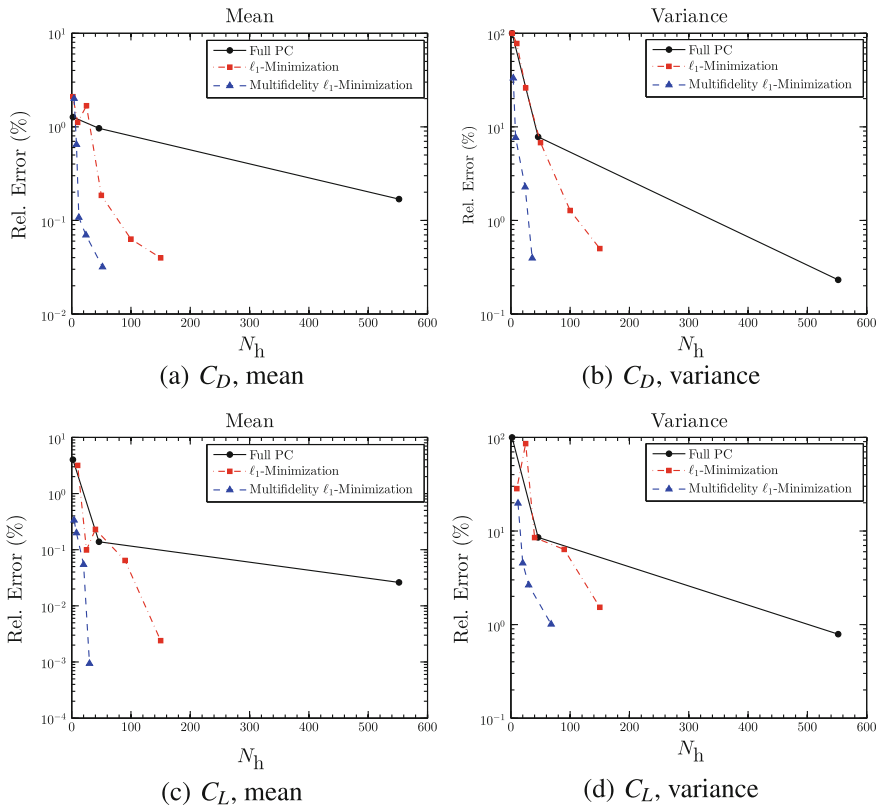


Fig. 5 Convergence of mean and variance of drag and lift coefficients for $\beta = 0.25$

error with $N_h = 100$, while the developed method needs only 20 samples. In this test case it was observed that the low-fidelity models are 20 times faster than high-fidelity. Assuming t is the required time for high-fidelity computations (fine grid), the total computational time for the developed method is $276t/20 + 20t \approx 34$. Therefore the developed method is almost 15 times and 3 times more efficient than the full PC and ℓ_1 -minimization methods, respectively.

Figure 6 presents the PCE coefficients of multifidelity ℓ_1 -minimization method recovered with 50 high-fidelity samples and $\beta = 0.25$. The coefficients are also compared to the reference coefficients (full PC with 552 samples). It should be reminded that the circles (blue and grays) show the reference PCE coefficients, while the multifidelity ℓ_1 -minimization coefficients are shown with the squares and triangles. It is observed that for both drag and lift coefficients, corrected coefficients (squares) are very similar to the reference coefficients and the multifidelity method

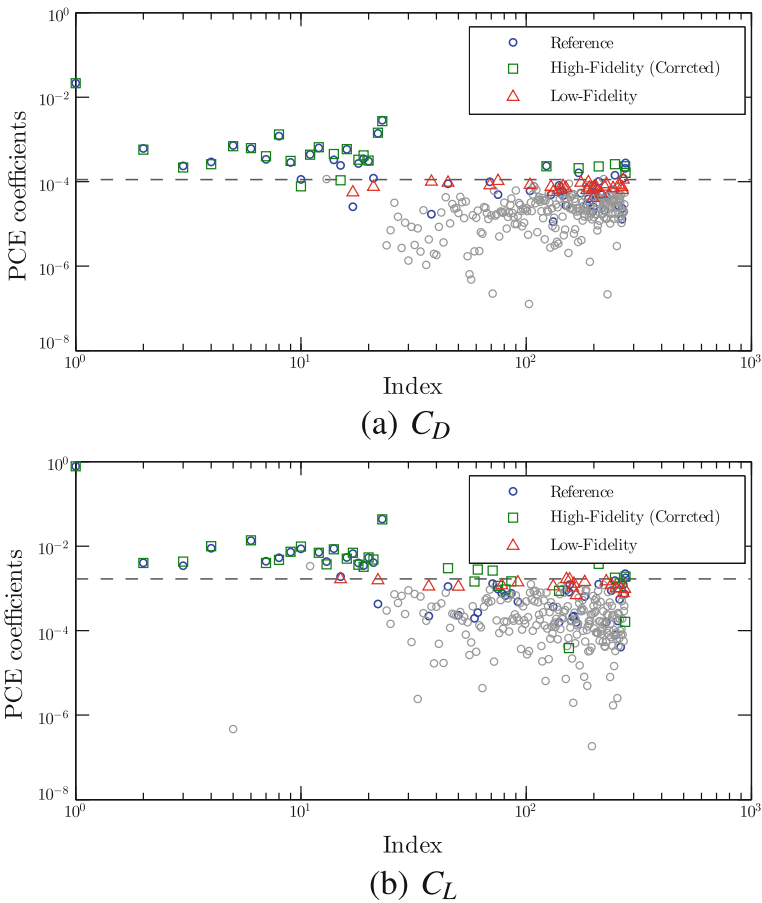
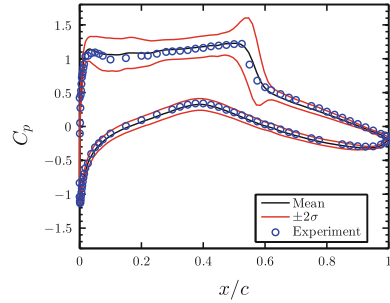


Fig. 6 Recovered PCE coefficients with $N_h = 50$

Fig. 7 C_p statistics computed with over-sampled full PC (reference solution)



has successfully corrected the low-fidelity PCE coefficients. In addition, it is clear that the coefficients with lower energy, that are calculated through inexpensive low-fidelity computations are slightly different with reference coefficients.

Figure 7 displays the mean and $\pm 2\sigma$ bounds of C_p distribution on the pressure and suction surfaces of airfoil using over-sampled full PC. This is again considered as the reference solution. In addition, ℓ_1 -minimization and multifidelity ℓ_1 -minimization results of C_p distribution with different number of high-fidelity samples are shown in Fig. 8. The comparison reveals that increasing number of samples improves both the C_p statistics returned by ℓ_1 -minimization and multifidelity ℓ_1 -minimization schemes. Furthermore, the developed method requires less samples than ℓ_1 -minimization to predict acceptable distributions. In addition, it is observed that an acceptable mean distribution can be obtained with the developed method using 10 samples. However, to achieve an accurate standard deviation distribution one needs to produce at least 20 samples, which is much less than other methods employed. This is due to the importance of high order terms contributions in the variance of output. One can see that the classical ℓ_1 -minimization cannot predict an accurate standard deviation distribution even with 50 samples.

Conclusion

In the present paper an efficient PCE methodology was introduced for efficient uncertainty quantification of sparse stochastic problems. The proposed method combines the advantages of ℓ_1 -minimization and multifidelity methods. The method first recovers the important modes of PCE using the ℓ_1 -minimization technique using low-fidelity computations. The low-fidelity computations are not supposed to yield precise QoI's, but they should be accurate enough to reflect the physical trends well. The multifidelity PCE method is then utilized to correct a subset of recovered coefficients using high-fidelity computations. The performance of the method is examined using the transonic flow around RAE2822 airfoil under operational and geometri-

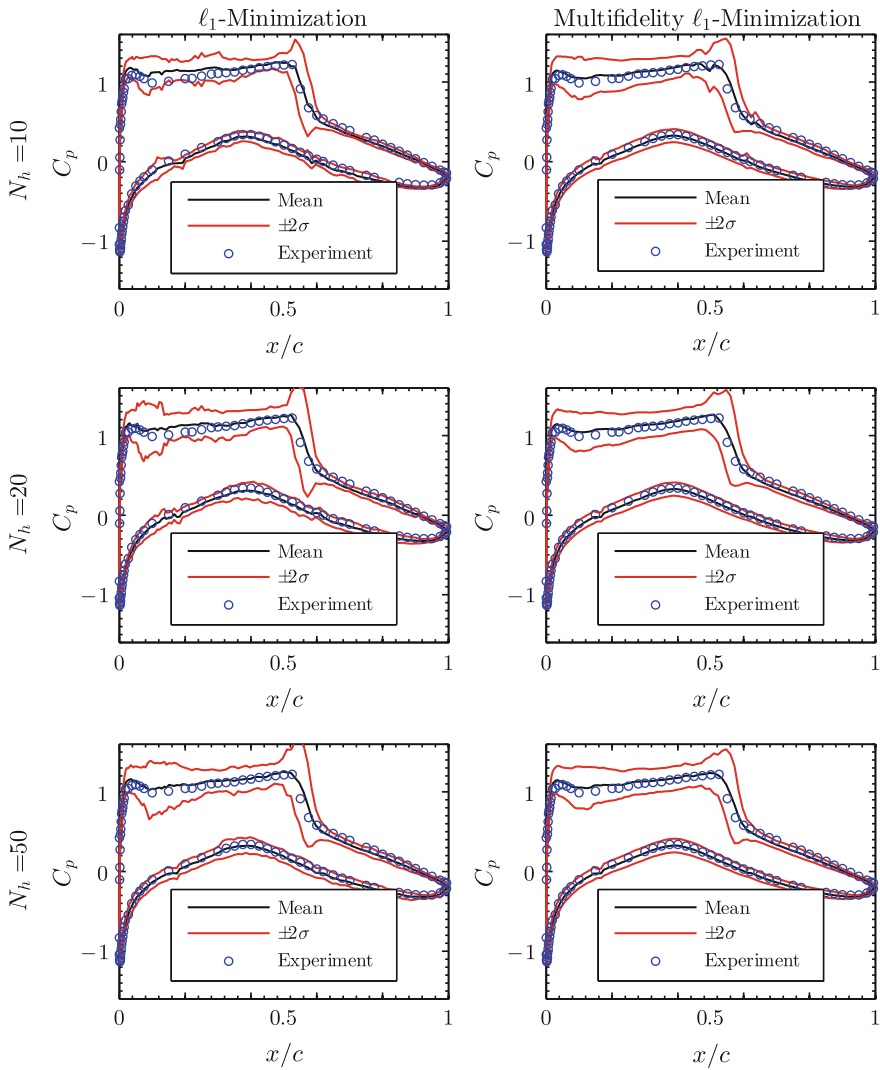


Fig. 8 Comparison of C_p statistics computed with ℓ_1 -minimization and multifidelity ℓ_1 -minimization methods

cal uncertainties. It was found that the developed method is more effective than the ℓ_1 -minimization and for the same accuracy the method needs lower number of high-fidelity samples. For the investigated test cases in this paper, it was observed that the present method is almost 15–20 times faster than the full PC method and 3–4 times faster than the classical ℓ_1 -minimization method.

References

- Cook P, Firmin M, McDonald M (1977) Aerofoil RAE 2822-Pressure distributions, and boundary layer and wake measurements. Experimental data base for computer program assessment, AGARD Report AR, p 138
- Davenport MA, Duarte MF, Eldar Y, Kutyniok G (2011) Introduction to compressed sensing. In: Kutyniok G, Eldar YC (eds) Compressed sensing, theory and applications. Cambridge University Press, pp 1–64
- Davis G, Mallat S, Avellaneda M (1997) Adaptive greedy approximations. *Constr. Approximation* 13(1):57–98
- Doostan A, Owhadi H (2011) A non-adapted sparse approximation of PDEs with stochastic inputs. *J Comput Phys* 230(8):3015–3034
- Fishman G (1996) Monte Carlo: concepts. Algorithms and applications, Springer, New York
- Ghanem R, Spanos P (1938) Stochastic finite elements a spectral approach, 3rd edn. Springer, New York
- Hosder S, Walters RW, Perez R (2006) A non-intrusive polynomial chaos method for uncertainty propagation in CFD simulations. In: 44th AIAA aerospace sciences meeting and exhibit
- Lucor D, Karniadakis GE (2004) Adaptive generalized polynomial chaos for nonlinear random oscillators. *SIAM J Sci Comput* 26(2):720–735
- Muthukrishnan S (2005) Data streams: algorithms and applications. Now Publishers Inc
- Nair PB, Keane AJ (2002) Stochastic reduced basis methods. *AIAA J* 40(8):1653–1664
- Needell D (2009) Topics in compressed sensing. Ph.D. thesis, University of California
- Ng LWT, Eldred M (2012a) Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA, p 1852
- Ng LWT, Eldred M (2012b) Multifidelity uncertainty quantification using nonintrusive polynomial chaos and stochastic collocation. In: Proceedings of the 14th AIAA non-deterministic approaches conference, number AIAA-2012-1852, vol 43. Honolulu, HI
- Pati YC, Rezaifar R, Krishnaprasad P (1993) Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: 1993 conference record of the twenty-seventh asilomar conference on signals, systems and computers, 1993, pp 40–44
- Salehi S, Raisee M, Cervantes MJ, Ahmad (2017) Efficient uncertainty quantification of stochastic CFD problems using sparse polynomial chaos and compressed sensing. *Comput Fluids* 154:296–321. <http://www.sciencedirect.com/science/article/pii/S0045793017302281>
- Sobol' I (1967) On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput Math Math Phys* 7(4):86–112
- Soize C, Ghanem R (2004) Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM J Sci Comput* 26(2):395–410
- Wan X, Karniadakis GE (2005) An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *J Comput Phys* 209(2):617–642
- Wiener N (1938) The homogeneous chaos. *Am J Math* 60(4):897–936
- Witteveen JA, Doostan A, Chantrasmith T, Pecnik R, Iaccarino G (2009) Comparison of stochastic collocation methods for uncertainty quantification of the transonic RAE 2822 airfoil. In: Proceedings of workshop on quantification of CFD uncertainties

Part V
Multi-disciplinary Design Optimization

Evolving Neural Networks to Optimize Material Usage in Blow Molded Containers



Roman Denysiuk, Fernando M. Duarte, João P. Nunes
and António Gaspar-Cunha

Abstract In industry, there is a growing interest to optimize the use of raw material in blow molded products. Commonly, the material in blow molded containers is optimized by dividing the container into different sections and minimizing the wall thickness of each section. The definition of discrete sections is limited by the shape of the container and can lead to suboptimal solutions. This study suggests determining the optimal thickness distribution for blow molded containers as a function of geometry. The proposed methodology relies on the use of neural networks and finite element analysis. Neural networks are stochastically evolved considering multiple objectives related to the optimization of material usage, such as cost and quality. Numerical simulations based on finite element analysis are used to evaluate the performance of the container with a thickness profile determined by feeding the coordinates of mesh elements in finite element model into the neural network. The proposed methodology was applied to the design of industrial bottle. The obtained results suggested the validity and usefulness of this methodology by revealing its ability to identify the most critical regions for the application of material.

Introduction

Blow molding is an important industrial processes for manufacturing hollow plastic parts. The production of jars, bottles and similar containers are among its main applications. Such products are widely used all over the world to contain liquids from

R. Denysiuk · F. M. Duarte · J. P. Nunes · A. Gaspar-Cunha (✉)
IPC - Institute for Polymer and Composites, University of Minho,
Campus de Azurém, Guimarães 4800-058, Portugal
e-mail: agc@dep.uminho.pt

R. Denysiuk
e-mail: roman@dep.uminho.pt

F. M. Duarte
e-mail: fduarte@dep.uminho.pt

J. P. Nunes
e-mail: jpn@dep.uminho.pt

drinks for human consumption to cosmetics and oil. In blow molding, a molten material is placed into a mold and inflated with gas whose pressure pushes the material out to match the mold. The costs of raw materials compose a significant share of the total costs of blow molded products. Thus, reducing costs and increasing competitiveness for manufacturing companies can be effectively achieved by minimizing the material use. This requires a trade-off between the costs of production and quality criteria, as reducing the amount of material can deteriorate important product properties.

The conventional trial-and-error approach is tedious and inefficient to optimize product development. It can lead to a significant waste of time and energy whereas the results are highly dependent on expert experience. Computer Aided Engineering (CAE) has become increasingly popular to support engineering tasks. Computer simulations and optimization can help to reduce the number of empirical trials, thus saving time and money. Numerical approaches such as Finite Element Methods (FEMs) and optimization techniques are promising and have a long history of use in blow molding design.

When optimizing blow molding, two major problems can be identified. The one is to determine a wall thickness distribution of the final container. The other is concerned with finding a shape of the preform and setting appropriate process parameters to produce a container with a desired thickness distribution. Optimization of thickness distribution is typically addressed by dividing the preform or container into distinct sections and optimizing the thickness of each section. In Laroche et al. (1999), the optimal preform thickness distribution was sought that yields a given uniform part thickness. In Gauvin et al. (2003), two approaches were presented, with the optimization aiming at finding a thickness distribution that minimizes the weight and satisfies mechanical constraints.

The other problem arising in blow molding is concerned with a process optimization aiming at finding the optimal operating conditions that minimize the weight and respect the thickness distribution found by the performance optimization. In Thibault et al. (2007), an approach to optimize the stretch blow molding process was presented, aiming at establishing the optimal preform geometry (thickness and shape) and optimal operating conditions to produce a container with a target thickness distribution. In the above studies, optimization was performed by gradient-based search methods. These methods have good theoretical properties and fast convergence. However, gradient-based methods are essentially local search techniques and their performance highly depends on the initial point.

Evolutionary algorithms (EAs) allow to overcome limitations associated with traditional optimization methods. EAs attempt to perform global search without using gradient information. In Huang and Huang (2007), genetic algorithm (GA) was used to find the optimal thickness distribution for preform. In Yang et al. (2014), particle swarm optimization was used to adjust parameters of a neural network in order to fit experimentally collected data and to obtain the appropriate lamp settings. The preform geometry was optimized in Biglione et al. (2016) to obtain a target wall thickness distribution. In Hopmann et al. (2015), this also included the optimization of process parameters.

A common feature that limits the applicability of the discussed approaches is that a container being optimized is divided into a number of sections, assuming a uniform thickness within each section. A proper division can be not straightforward as it greatly depends on the geometry of container. Poor results can be obtained if sections are inadequately defined. Also, such approach can lead to discontinuities in junctions between sections.

The specific contributions of this paper are the application of a regression model to find the optimal thickness distribution as a function of the container's geometry and solving the problem using multiobjective neuroevolutionary algorithm.

Problem Formulation

This study aims at developing a methodology for the optimization of material usage in blow molded containers, which is also a major concern for industry due to the influence of the costs of raw materials on the total production costs. The particular industrial bottle whose design is herein addressed has a diameter of 395 mm and a height of 625 mm. The material is plastic with the mass density of $1.15 \times 10^{-9} \text{ g/cm}^3$ and Poisson's ration of 0.4. The bottle is set to experience a blowing pressure. The ratio between the pressure and Young's modulus is 0.0014. The minimum and maximum allowable values of wall thickness are 0.1 and 2 mm, respectively. Figure 1 shows the geometry model of the bottle used in this study.

The problem consists in determining the optimal wall thickness distribution. This problem involves several criteria that must be considered, such as the cost of utilized

Fig. 1 Geometry model of the bottle



material and the product quality. In order to capture possible trade-offs between these criteria, the problem is posed as a multiobjective optimization problem. The mass of the container and the stress are two objectives (f_1 and f_2) to be minimized. These objectives are estimated by computer simulations performed by Abaqus, the finite element analysis software.

Neuroevolutionary Thickness Optimization

This section describes the proposed methodology to design blow molded containers with an optimized wall thickness distribution. The main idea consists in treating the wall thickness as a function of container's geometry. The realization of this ideal relies on capabilities of neural networks. Neural networks are used to convert the coordinates along the wall into the thickness values. This can also be viewed as a regression model. Though, it is important to point out the distinction between a traditional regression that makes use of data points with known input and target variables and the proposed methodology where only input variables are available. In turn, this hinders the application of traditional gradient-based methods to learn the parameters of the neural network. To overcome this issue, neuroevolution is used. Neuroevolution refers to the use of evolutionary algorithms to evolve neural networks. It provides the potential to evolve both the topology and parameters of neural networks. The outline of the proposed neuroevolution is given by Algorithm 1.

First, a population of neural networks is randomly generated in the initialization procedure. Each individual in the population is represented by two chromosomes. The first is defined by a binary string that determines the network topology by indicating which neurons are used in the hidden layer. The second is given by a real-valued string that encodes all weights and biases in the neural network.

Each time a new individual is generated it is sent for evaluation. The evaluation procedure comprises decoding the individual's genotype into the neural network, calculating the thickness profile of the bottle and computing the objective values reflecting its performance.

Figure 2 graphically illustrates the idea behind the calculation of the thickness profile. The coordinates of each mesh element in finite element model are fed into the neural network. The output is the thickness at the corresponding location. Processing this way all the mesh elements gives a thickness profile of the bottle. The resulting finite element model with the calculated thickness profile is submitted to perform computer simulation, whose subsequent output is read to extract the values of the mass and stress.

The population of neural networks is evolved for a predefined number of generations using a steady-state variant of evolutionary process (lines 2–6 in Algorithm 1). This means a single offspring is produced in each generation. Selection aims at selecting parents for producing offspring. This study uses a simple uniform selection where each population member has an equal chance to be selected.

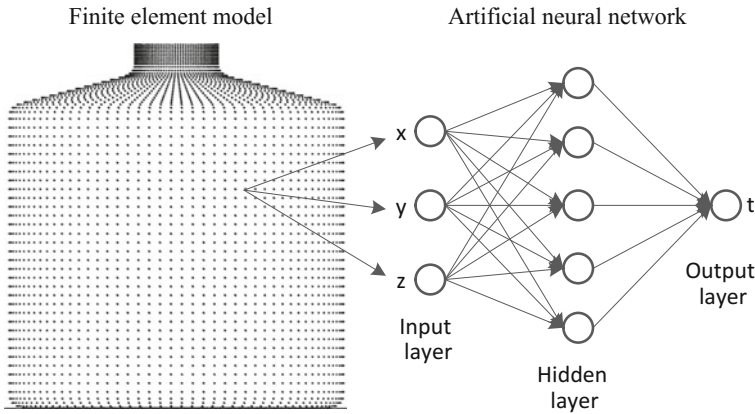


Fig. 2 Thickness calculation

Algorithm 1 Neuroevolution	
1:	initialization()
2:	repeat
3:	selection()
4:	variation()
5:	replacement()
6:	until the stopping criterion is met

Evolutionary operators are applied to parents in order to produce offspring in the variation procedure. Variation plays a crucial role in the exploration of the search space. Multichromosomal representation used in this study allows the application of operators that proved effective in the exploration of binary and continuous search spaces. Herein, a simple bit-flip mutation is performed on the binary string. A particular attention is given to a continuous variation operator, as most of the genome is represented by a real-valued chromosome encoding all weights and biases, which are central to the expressiveness of the neural network. Different continuous variation operators are investigated in the experimental study.

Replacement aims at forming a population of the next generation relying on the concept of the survival of the fittest from natural evolution. As the proposed neuroevolution is designed to deal with multiple objectives, replacement must ensure the convergence and diversity of population. These two requirements are known to be somewhat conflicting in nature. The adopted replacement strategy relies on the concept of the Pareto dominance to provide convergence and the hypervolume measure to ensure diversity. First, the population is sorted using the nondominated sorting procedure to find individuals in the last nondominated front. Then among the found individuals the one with the least volume of exclusively dominated objective space is removed (Beumea et al. 2007).

The result of the above described process is expected to be a set of neural networks, where each neural network gives the design of the container providing a specific trade-off between its mass and mechanical properties.

Computational Experiments

Experimental Setup

Neuroevolutionary algorithm developed for optimizing a wall thickness distribution was investigated with different variation operators, taking advantage of multichromosomal representation. For real-coded genetic algorithm (GA) operator, simulated binary crossover (SBX) and polynomial mutation (PM) were used. Evolution strategy (ES) operator was used with a non-isotropic mutation. Differential evolution (DE) operator with *rand/1/bin* variant and PM was employed. Covariance matrix adaptation (CMA) operator was used with a step size adaptation rule relying on a population-based notion of success.

The numerical simulations based on finite element analysis were carried out by Abaqus 6.13-4 provided in a network licensing format. A high computational time required for each simulation and a limited number of analysis jobs allowed to be run simultaneously on a network restricted considerably optimization runs. Thus, five independent runs were performed by each neuroevolutionary variant. The population size of 50 was used and the number of available evaluations was 100. The other parameter settings are shown in Table 1. The results were quantitatively assessed using the hypervolume measure (Zitzler and Thiele 1998).

Table 1 Parameter settings (n —is a chromosome length)

Operator	Parameters
SBX	$p_c = 1, \eta_c = 20$
PM	$p_m = 1/n, \eta_m = 20$
ES	$\tau_0 = 1/\sqrt{2n}, \tau_1 = 1/\sqrt{2\sqrt{n}}, \sigma_0 = \sqrt{1/(3n)}$
DE	$CR = 1, F = 0.5$
CMA	$d = 1 + n/2, p_{succ}^{target} = 1/(5 + \sqrt{1/2}),$ $c_p = p_{succ}^{target} / (2 + p_{succ}^{target}), c_c = 2/(n + 2),$ $c_{cov} = 2/(n^2 + 6), p_{thresh} = 0.44$

Results

Table 2 shows the final results of statistical runs with respect to the hypervolume. The last row of the table refers to the hypervolume for all nondominated solutions obtained by combining results of the five runs. The results indicate that GA is the worst performing operator. It can be because crossover treat genes independently when producing offspring. This causes a disruptive effect on the linkage between genes. Although ES does not explicitly accounts for relations between genes, its slightly better performance can be explained by the self-adaption mechanism that learns mutation strength for each gene. Both DE and CMA mechanisms allow for the adaptation to fitness landscape. CMA works better for extreme runs, whereas DE gives the best median value and the hypervolume for the approximation set composed by the results of all runs. This can be because CMA has a larger number of parameters that require proper settings.

Since the hypervolume values presented in Table 2 differ slightly, it can misleadingly appear that the results obtained by different variants are quite similar. However, such seemingly small differences with respect to the hypervolume can be significant in practical terms. It can be understood when comparing the results of GA and DE variants, whose Pareto front approximations are shown in Fig. 3. Solutions forming both approximations lie sufficiently close to each other, being the most distant in the vicinity of the knee points.

However, the knee point region of the Pareto front is particularly interesting from an engineering perspective. When comparing the results with respect to the hypervolume DE variant gives an improvement of 1.71% relative to GA. Whereas in terms of the knee solutions the reduction by 32.34% of the material usage is achieved, which is significant for industry.

Visualization of the obtained Pareto optimal solutions leads to several important observations. In particular, there is a part of the Pareto optimal region where the material usage can be significantly reduced from the maxim value of 2.032 kg to approximately 0.2 kg with relatively a small degradation in mechanical properties. This can be the most interesting part from a practical perspective. However, a further reduction in the material results in a significant degradation of mechanical properties. Although such solutions are appealing from an economic point of view, they may be unacceptable as important quality criteria can be not met. Thus, these results further highlight the importance of proper tools to support product development.

Table 2 Results for statistical runs of different operators

Hypervolume	GA	ES	DE	CMA
Min.	0.9687	0.9713	0.9805	0.9882
Median	0.9701	0.9723	0.9897	0.9893
Max.	0.9746	0.975	0.9902	0.9905
Total	0.9749	0.976	0.9918	0.9916

Fig. 3 Nondominated solutions

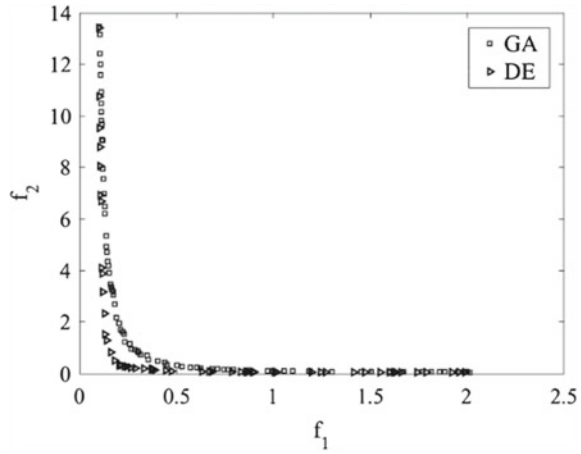


Fig. 4 Nondominated solutions near the knee points

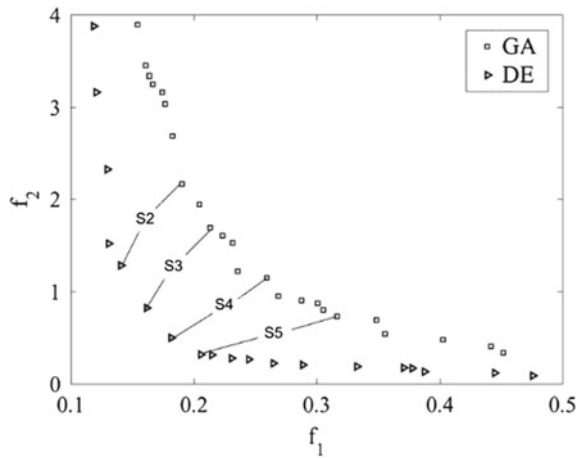


Figure 4 shows the Pareto optimal solutions in the vicinity of the knee points. For both approximations, four different solutions are highlighted. For these solutions and solutions S1 and S6 that refer to the corners of the Pareto fronts, Fig. 5 depicts the wall thickness of bottle along the vertical axis from top to down. Corner solutions S1 and S6 represent extreme scenarios with minimum and maximum values of thickness along the entire bottle. On the other hand, solutions S2–S5 show intermediate scenarios representing different trade-offs between the use of material and mechanical properties. The difference can be observed between the thickness distributions of solutions given by GA and DE. The former offers larger values starting from the top. Whereas the latter yields values close to the minimum for the most positions and only increases thickness in the bottom of bottle. This way, neuroevolution identifies the most critical regions for applying material. It is also noteworthy that GA provides a valid design of the bottle. The comparison is for illustration purpose and to stress

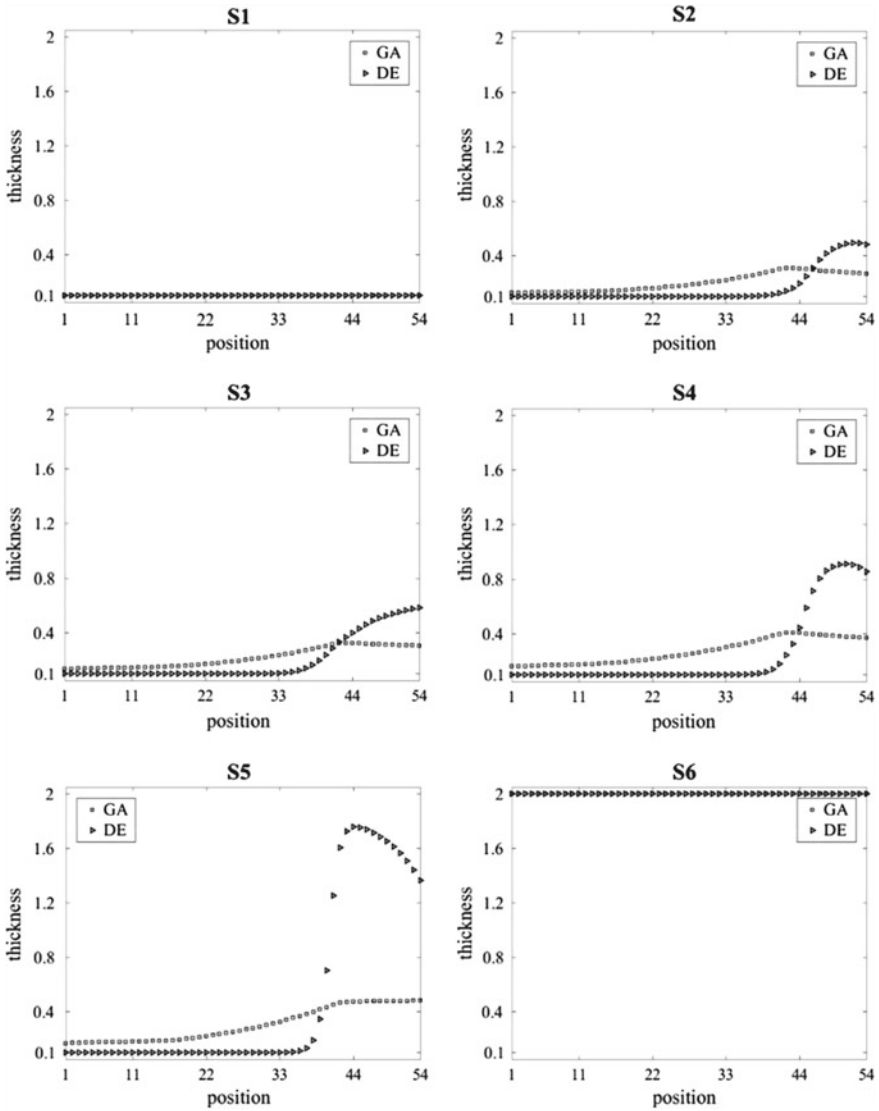


Fig. 5 Thickness distribution from top to bottom of the bottle

the importance of the neuroevolutionary design. Overall, these results demonstrate the ability of the proposed methodology to determine the material distribution for the bottle given its geometry, characteristics and design criteria to meet.

Conclusions

In blow molding industry, the product competitiveness can be effectively increased by reducing the costs of raw materials. This study suggested a methodology to optimize the material usage in blow molded products. This methodology aims at determining the optimal distribution of material as a function of the product geometry. Motivated by the universal approximation property, this function is approximated by neural network. The structure and parameters of the network are determined by neuroevolution. The search is performed addressing multiple objectives, minimizing the usage of material and the degradation of mechanical properties. This leads to a set of Pareto optimal networks representing different trade-offs between the objectives, which allows to obtain a valuable information about design alternatives and enables a posteriori decision making.

The application of the proposed methodology is demonstrated in a case study addressing the design of industrial bottle. Finite element analysis software was employed to simulate the response of the particular design to a static pressure. Different variants of neuroevolutionary algorithm were investigated. The obtained results indicate the importance of using proper search strategies and the ability of neuroevolution to optimize the thickness distribution under given conditions. Generality is a major advantage of the proposed methodology, as its applicability is independent of the bottle geometry.

In future, the developed methodology will be applied to optimize the preform and operating conditions.

Acknowledgements This work has been supported by FCT—Fundação para a Ciência e Tecnologia in the scope of the project: PEst-OE/EEI/UI0319/2014 and the European project MSCA-RISE-2015, NEWEX, with reference 734205.

References

- Beumea N, Naujoksa B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur J Oper Res* 181(3):1653–1669
- Biglione J, Béreaux Y, Charneau J-Y, Balcaen J, Chhay S (2016) Numerical simulation and optimization of the injection blow molding of polypropylene bottles—a single stage process. *Int J Mater Form* 9:471–487
- Gauvin C, Thibault F, Laroche D (2003) Optimization of blow molded part performance through process simulation. *Polym Eng Sci* 43:1407–1414
- Hopmann Ch, Rasche S, Windeck C (2015) Simulative design and process optimization of the two-stage stretch-blow molding process. *AIP Conf Proc* 1664:050011
- Huang G-Q, Huang H-X (2007) Optimizing parison thickness for extrusion blow molding by hybrid method. *J Mater Process Technol* 182:512–518
- Laroche D, Kabanemi KK, Pecora L, DiRaddo RW (1999) Integrated numerical modeling of the blow molding process. *Polym Eng Sci* 39:1223–1233
- Thibault F, Malo A, Lancot B, DiRaddo R (2007) Preform shape and operating condition optimization for the stretch blow molding process. *Polym Eng Sci* 47(3):289–301

- Yang Z, Naeem W, Menary G, Deng J, Li K (2014) Advanced modelling and optimization of infrared oven in injection stretch blow-moulding for energy saving. In: Proceedings of World Congress IFAC, pp 766–771
- Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms—A comparative case study. In: Proceedings of the 13th international conference on parallel problem solving from nature, pp 292–304

Coupled Subsystem Optimization for Preliminary Core Engine Design



Simon Extra, Michael Lockan, Dieter Bestle and Peter Flassig

Abstract In this paper we present an approach for preliminary aero engine design with the overall goal to improve the core engine with regards to efficiency and emissions. This is achieved through the usage of a cascaded multi-criterion global optimization strategy, where the overall optimization is performed on the basis of decoupled black box sub-optimization of components. Such an approach has the advantage to cut down the number of design variables for each optimization problem involved and to use legacy code for components.

Introduction

Due to the high complexity of aero engines and ever rising requirements regarding efficiency and emissions, the application of optimization methods represents a reasonable way to further increase engine performance. The system complexity, however, typically prohibits an all-at-once approach because of the high number of design variables and dependencies between the subsystems. Here, a solution to this is presented in the style of collaborative optimization which coordinates the interaction of subsystem optimization processes. This allows an optimization of the whole core engine while employing dedicated, subsystem-specific tools for each component

S. Extra (✉) · M. Lockan · D. Bestle

Engineering Mechanics and Vehicle Dynamics, Brandenburg University of Technology
Cottbus-Senftenberg, Siemens-Halske-Ring 14, 03046 Cottbus, Germany
e-mail: extra@b-tu.de

M. Lockan
e-mail: michael.lockan@b-tu.de

D. Bestle
e-mail: bestle@b-tu.de

P. Flassig
Rolls-Royce Deutschland Ltd. & Co KG,
Eschenweg 11, 15827 Blankenfelde-Mahlow, Germany
e-mail: Peter.Flassig@Rolls-Royce.com

analysis. A first step towards collaborative optimization was made by Sobieszczanski-Sobieski (1989) resulting in Concurrent Subspace Optimization (CSSO). It is based on the Global Sensitivity Equations (GSE) by Adelman and Haftka (1986) which allow to approximate the influence of a local parameter change on the global system behavior. Thereby it is possible to firstly optimize the subsystems with regards to local and shared design variables while still observing the global coupling values. The system coordination is performed in a subsequent step which is required to match all shared design variables, because they may change during subsystem optimizations.

The CSSO method was extended to employ response surfaces by Sellar et al. (1996) improving convergence for complex systems. A method without the use of a system coordination step was presented by Sobieszczanski-Sobieski et al. (1998). Using the BLISS (Bi-Level Integrated System Synthesis) method, the shared variables between subsystems are prescribed and kept constant during subsystem optimization, which is why no coordination step is needed.

Finally Sobieszczanski-Sobieski et al. (2003) proposed a strategy without the need for global sensitivity equations, as they are rather expensive to calculate. This concept, named BLISS-2000, prescribes the coupling parameters as well as the shared design variables for all subsystem optimizations during the process. The subsystem responses are used as a subsystem objective and weighted in a sum, where the weight is prescribed by the system optimizer as a design variable. However, each system analysis requires optimization of all subsystems, and thus the number of global calculation steps is rather limited.

In this paper, a strategy similar to BLISS-2000 is used due to the fact that all coupling parameters can be kept constant during subsystem optimization and no sensitivities need to be calculated. As the application is quite specific in contrast to the universal BLISS-2000 algorithm, some modifications can be made. More precisely, the output of each subsystem is prescribed as a fixed constraint by the global optimizer and has to be fulfilled as a local constraint or used as an input value. This strategy also allows for using legacy code in the subsystem processes, which is very useful in the context of engine design presented in this paper.

Core Engine Optimization

The core engine of an aero engine typically consists of compressor (C), combustor (B) and turbine (T). The objectives of its global optimization are improved engine efficiency as well as reduced emissions. The global efficiency may be defined as product of the three component efficiencies, i.e.,

$$\eta(\mathbf{p}) = \eta_C(\mathbf{p}) \eta_B(\mathbf{p}) \eta_T(\mathbf{p}) \quad (1)$$

depending on the overall design vector \mathbf{p} . Emissions are expressed by the emission index E_I which is defined by the ICAO (2011) as mass of NO_x , CO and

UHC (unburned hydrocarbon) in grams generated per kilogram of burned fuel, and reads as

$$E_I(\mathbf{p}) = \frac{m_{NO_x} + m_{CO} + m_{UHC}}{m_{Fuel}} . \quad (2)$$

The emission index is mainly determined by the performance of the combustion chamber, but the overall configuration of the core engine influences the result as well. This is due to the fact that the performance of the combustor is influenced by the arrangement of the Joule cycle.

The global optimization problem for the whole, coupled core engine process may be defined as

$$\min_{\mathbf{p} \in P} \begin{bmatrix} -\eta \\ E_I \end{bmatrix} \text{ s.t. } P = \{ \mathbf{p} \in \mathbb{R}^d \mid \mathbf{h}(\mathbf{p}) \leq \mathbf{0}, \mathbf{p}^l \leq \mathbf{p} \leq \mathbf{p}^u \} \quad (3)$$

where the design vector \mathbf{p} of dimension d summarizes the local component design parameters $\mathbf{p}_C, \mathbf{p}_B, \mathbf{p}_T$ as well as some globally shared design variables \mathbf{p}_S :

$$\mathbf{p} = [\mathbf{p}_C^T, \mathbf{p}_B^T, \mathbf{p}_T^T, \mathbf{p}_S^T]^T . \quad (4)$$

Also the constraint vector \mathbf{h} summarises the constraints for each subprocess as

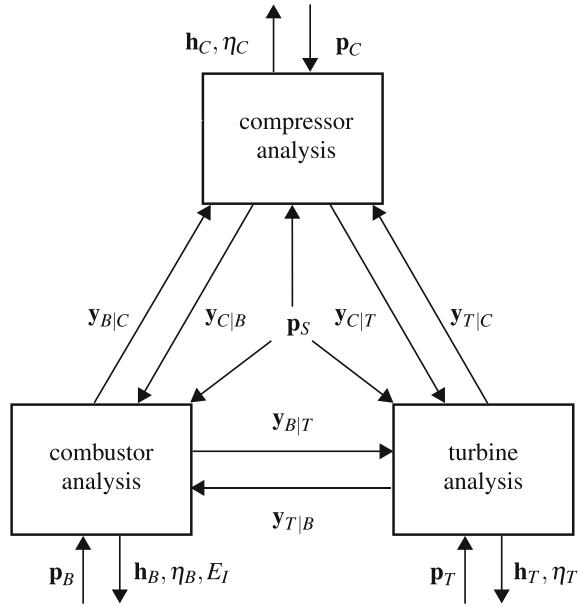
$$\mathbf{h} = [\mathbf{h}_C^T, \mathbf{h}_B^T, \mathbf{h}_T^T]^T . \quad (5)$$

Figure 1 shows all subprocesses as currently employed in the design of a core engine, as well as the couplings. Each of these subprocesses returns a part of the objectives (1), (2) and constraints (5). Further it returns some coupling states $\mathbf{y}_{i|j}$ which are functions of the shared design variables \mathbf{p}_S , the respective local design variables as well as coupling variables defined by other components, respectively. The dependencies for compressor, combustor and turbine may be formalized as

$$\begin{aligned} \mathbf{y}_C &= [\mathbf{h}_C^T, \eta_C, \mathbf{y}_{C|B}^T, \mathbf{y}_{C|T}^T]^T = \mathbf{f}_C(\mathbf{p}_C, \mathbf{p}_S, \mathbf{y}_{B|C}, \mathbf{y}_{T|C}) , \\ \mathbf{y}_B &= [\mathbf{h}_B^T, \eta_B, E_I, \mathbf{y}_{B|C}^T, \mathbf{y}_{B|T}^T]^T = \mathbf{f}_B(\mathbf{p}_B, \mathbf{p}_S, \mathbf{y}_{C|B}, \mathbf{y}_{T|B}) , \\ \mathbf{y}_T &= [\mathbf{h}_T^T, \eta_T, \mathbf{y}_{T|C}^T, \mathbf{y}_{T|B}^T]^T = \mathbf{f}_T(\mathbf{p}_T, \mathbf{p}_S, \mathbf{y}_{C|T}, \mathbf{y}_{B|T}) . \end{aligned} \quad (6)$$

As we can see from these equations, the coupled problem for the optimization of a core engine is highly complex due to the dependencies among the components and the amount of variables involved, where an improved configuration cannot reliably be found. This leads to the idea of decomposition, i.e., to use decoupled component optimization problems, which facilitates the use of legacy code for the subprocesses and reduces the amount of design variables for optimization. A global optimizer then

Fig. 1 Scheme of a three-component, non-hierarchical, coupled core engine design problem



has to enforce the synchronization of subprocess results via prescribing values for all \mathbf{p}_S and \mathbf{y}_{ij} to be fulfilled in subsystem optimization tasks.

Tasks of the Global Optimization Process

In this section the subprocesses used in the decoupled core engine optimization as well as the global synchronisation problem are presented. Let

$$\mathbf{s} = \left[\tilde{\mathbf{y}}_{C|B}^T, \tilde{\mathbf{y}}_{C|T}^T, \tilde{\mathbf{y}}_{B|C}^T, \tilde{\mathbf{y}}_{B|T}^T, \tilde{\mathbf{y}}_{T|C}^T, \tilde{\mathbf{y}}_{T|B}^T, \mathbf{p}_S^T \right]^T \tag{7}$$

summarize all coupling vectors for the global optimization problem and shared design variables \mathbf{p}_S . All prescribed interface couplings are marked with a tilde ($\tilde{\bullet}$). In the current state-of-the-art approach, decoupled subsystem optimization processes are used with fixed interface vectors which represents a much more manageable problem compared to the all-at-once approach in Eq. (3). Each fixed interface vector is defined by a previously executed, lower fidelity calculation performed in a preceding step of the design workflow. Whilst this allows for an independent optimization of each component by a specialized group of experts, the interface vectors do not inevitably represent the optimal coupling configuration for the optimized subsystems. Therefore, they need to be adapted in an overall system optimization (11), (12), however, they may be considered as given in the component optimization subprocesses.

Compressor Subprocess

The optimization subproblem for the compressor depends on global variables, local design variables and local constraints. Only the local design variables \mathbf{p}_C are changed to improve the component-specific objective, i.e., to maximize compressor efficiency η_C , to fulfil component-specific constraints $\mathbf{h}_C \leq \mathbf{0}$ and to come up to the prescribed values (7) with its output states $\mathbf{y}_{C|B}$ and $\mathbf{y}_{C|T}$. A tolerance ε_C is provided for these coupling constraints due to the fact that the local optimization may not be able to really converge to arbitrary prescribed values, but it enforces the algorithm to match results with some precision:

$$\begin{aligned} \min_{\mathbf{p}_C \in P_C} (-\eta_C) \text{ s.t.} \\ P_C = \{ \mathbf{p}_C \in \mathbb{R}^{d_C} \mid \mathbf{h}_C(\mathbf{p}_C) \leq \mathbf{0}, \|\mathbf{y}_{C|B}(\mathbf{p}_C) - \tilde{\mathbf{y}}_{C|B}\| \leq \varepsilon_C, \\ \|\mathbf{y}_{C|T}(\mathbf{p}_C) - \tilde{\mathbf{y}}_{C|T}\| \leq \varepsilon_C, \mathbf{p}_C^l \leq \mathbf{p}_C \leq \mathbf{p}_C^u \} . \end{aligned} \quad (8)$$

The values for the input coupling states as well as the shared design variables are fixed in this context, i.e., $\tilde{\mathbf{y}}_{B|C}, \tilde{\mathbf{y}}_{T|C}, \mathbf{p}_S = \text{const}$. The optimization subprocess returns an optimal compressor efficiency η_C^* subject to the local variables for the given global configuration as prescribed by the shared design variables and coupling variables. In contrast to Eq. (6) there is no return of the output coupling values, as they are prescribed by the global optimizer and are used as local inequality constraints.

Combustor Subprocess

The optimization problem for the combustor subprocess is defined analogously as

$$\begin{aligned} \min_{\mathbf{p}_B \in P_B} (wE_I - (1-w)\eta_B) \text{ s.t.} \\ P_B = \{ \mathbf{p}_B \in \mathbb{R}^{d_B} \mid \mathbf{h}_B(\mathbf{p}_B) \leq \mathbf{0}, \|\mathbf{y}_{B|C}(\mathbf{p}_B) - \tilde{\mathbf{y}}_{B|C}\| \leq \varepsilon_B, \\ \|\mathbf{y}_{B|T}(\mathbf{p}_B) - \tilde{\mathbf{y}}_{B|T}\| \leq \varepsilon_B, \mathbf{p}_B^l \leq \mathbf{p}_B \leq \mathbf{p}_B^u \} . \end{aligned} \quad (9)$$

Here we have two optimization goals which are combined by a weighted sum approach. The weighting factor w is prescribed for the local combustor optimization process, but may be handled as a design variable by the global optimization process. The local design vector \mathbf{p}_B is varied to meet the local constraints $\mathbf{h}_B \leq \mathbf{0}$ as well as the prescribed output coupling parameters $\tilde{\mathbf{y}}_{B|C}$ and $\tilde{\mathbf{y}}_{B|T}$. The input coupling parameters, shared design variables and w are held constant as $\tilde{\mathbf{y}}_{C|B}, \tilde{\mathbf{y}}_{T|B}, \mathbf{p}_S, w = \text{const}$. during local optimization. The return values are determined by the weighted objective of the combustor subprocess, and the optimal values of η_B^* and E_I^* are used in the overall optimization problem for the global improvement of efficiency and emission index.

Turbine Subprocess

The turbine subsystem optimization process is again a single-objective optimization maximizing the turbine efficiency:

$$\begin{aligned} & \min_{\mathbf{p}_T \in P_T} (-\eta_T) \text{ s.t.} \\ & P_T = \{ \mathbf{p}_T \in \mathbb{R}^{d_T} \mid \mathbf{h}_T(\mathbf{p}_T) \leq \mathbf{0}, \|\mathbf{y}_{T|C}(\mathbf{p}_T) - \tilde{\mathbf{y}}_{T|C}\| \leq \varepsilon_T, \\ & \quad \|\mathbf{y}_{T|B}(\mathbf{p}_T) - \tilde{\mathbf{y}}_{T|B}\| \leq \varepsilon_T, \mathbf{p}_T^l \leq \mathbf{p}_T \leq \mathbf{p}_T^u \} \end{aligned} \quad (10)$$

where the local problem is optimized subject to fixed coupling vectors from compressor and combustor as well as a constant shared design vector \mathbf{p}_S prescribed by the global optimizer, i.e., $\tilde{\mathbf{y}}_{C|T}, \tilde{\mathbf{y}}_{B|T}, \mathbf{p}_S = \text{const}$. The return of the optimization process is the optimal turbine efficiency η_T^* achieved through changes of the local design variables \mathbf{p}_T to fulfil local constraints $\mathbf{h}_T \leq \mathbf{0}$ subject to global constraints.

System Optimization Process

The decoupled component optimization as presented above severely limits the amount of improvement of the design goals (1), (2) due to fixed interface states. To change this, a global system optimizer for the whole core engine is introduced which can be scaled to any number of subsystems. It is applied to a reduced subsystem in the next section. The optimizer improves the core engine performance through variations of both interface couplings and shared design variables, whilst allowing the subsystem optimizations to find their own optima with respect to the prescribed values. This enables global improvement whilst still keeping the advantages of a decoupled process, i.e., the reduced number of actually considered design variables compared to an all-at-once strategy. Each step of the global optimization involves the same amount of calculations as the core engine optimization discussed at the beginning of this section. The optimal configuration of the core engine can be found through an unconstrained, multi-objective global optimization maximizing the core engine efficiency and minimizing the emission index as defined in Eqs. (1) and (2), respectively:

$$\min_{\mathbf{s}^w \in S} \begin{bmatrix} -\eta^* \\ E_I^* \end{bmatrix} \text{ s.t. } S = \{ \mathbf{s}^w \in \mathbb{R}^{d_S} \mid \mathbf{s}^{w,l} \leq \mathbf{s}^w \leq \mathbf{s}^{w,u} \} \quad (11)$$

where the global efficiency is

$$\eta^* = \eta_C^* \eta_B^* \eta_T^* , \quad (12)$$

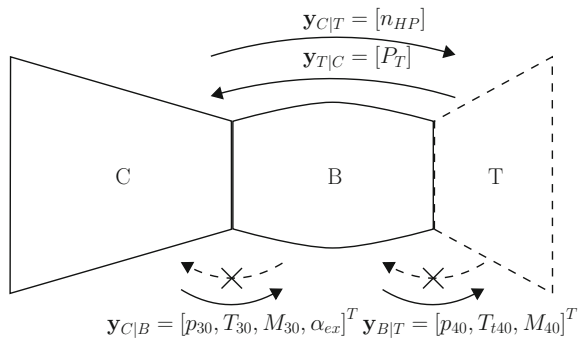
and the global parameter vector $\mathbf{s}^w = [\mathbf{s}^T, w]^T$ contains the global coupling and shared design variable vector \mathbf{s} as well as the weighting factor w used in the combustor subprocess (9). This is a deviation from the BLISS-2000 strategy as no weight is applied on the single objective compressor subprocess, which is permissible because any improvement in compressor efficiency always contributes to an improvement in overall efficiency. In contrast to (3), no component design parameters are observed here, only global variables are influenced by the global optimizer. Both optimization objectives are built up from results of subsystem optimizations denoted by the asterisk (\bullet^*). The complexity of the optimization problem in Eq. (11) shows the need for an efficient global optimization strategy. The use of a general purpose multi-objective genetic algorithm would require a very high amount of subsystem optimizations and thus lead to a very long computation time. To circumvent these problems, two measures are taken: firstly, Eq. (11) is transferred into a scalar, weighted sum problem with a fixed weight w^S to find specific Pareto-optimal tradeoffs according to pre-defined priorities, and secondly an efficient response surface algorithm is used to solve this problem as outlined in the following section.

Solution Concept and Application Example

In order to investigate a proper solution concept in reasonable time, the optimization problem (8)–(12) is stripped down to two components only. These are the compressor with design problem (8) and the combustor with design problem (9), whereas the turbine remains unchanged, i.e., $\mathbf{p}_T = const$. Also the considered couplings are restricted according to Fig. 2. We assume no upstream influence $\tilde{\mathbf{y}}_{B|C} = []$ from combustor to compressor and $\tilde{\mathbf{y}}_{T|B} = []$ from turbine on combustor, and the upstream coupling from turbine on combustor by provided turbine power P_T is assumed to be prescribed, i.e.,

$$\mathbf{y}_{T|C} = [P_T] := \tilde{\mathbf{y}}_{T|C} = const. \tag{13}$$

Fig. 2 Application example



Downstream compressor-combustor coupling

$$\mathbf{y}_{C|B} = [p_{30}, T_{30}, M_{30}, \alpha_{ex}]^T \quad (14)$$

by static pressure p_{30} , temperature T_{30} , Mach number M_{30} and compressor flow exit angle α_{ex} in the radial plane is assumed. The downstream coupling from combustor to turbine is prescribed as

$$\mathbf{y}_{B|T} = [p_{40}, T_{40}, M_{40}]^T := \tilde{\mathbf{y}}_{B|T} = \text{const.} \quad (15)$$

summarizing the static pressure p_{40} , the total temperature T_{40} and Mach number M_{40} . The coupling

$$\mathbf{y}_{C|T} = [n_{HP}] := \tilde{\mathbf{y}}_{C|T} = \text{const.} \quad (16)$$

from compressor on turbine by high pressure shaft speed n_{HP} is also prescribed. For optimization, the local compressor design variable \mathbf{p}_C and combustor design variables \mathbf{p}_B are modified. Further the shared design parameters

$$\mathbf{p}_S = [l_C, h_p, r_p]^T \quad (17)$$

summarizing compressor length l_C as well as mean height h_p and mean radius r_p of the gas path at compressor exit are adapted. The compressor length is a shared design variable, since we assume a prescribed total length of the whole core engine where an elongated compressor will reduce the axial length of combustor and turbine. In this application only the length of compressor and combustor will be varied. The geometric design parameters h_p and r_p are shared between compressor exit and combustor inlet.

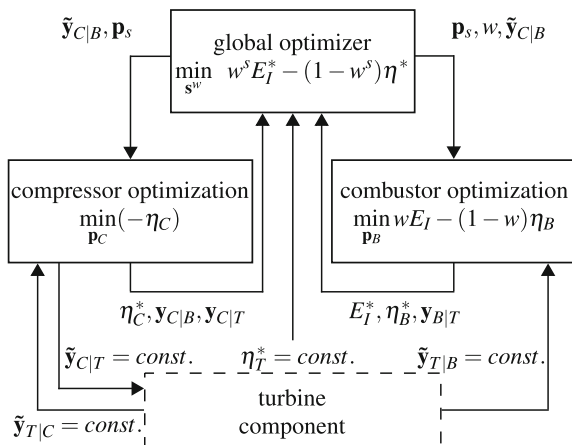
Figure 3 shows a scheme of the coupled optimization (8)–(12). Due to the simplifications, the turbine is considered only as a constant representation of the reference design providing the turbine efficiency η_T^* according to the prescribed coupling and shared design variables. The global optimizer minimizes the emission index E_I^* provided by the combustor sub-optimization and maximises the overall efficiency η^* where η_C^* and η_B^* result from compressor and combustor optimization, respectively. For simplicity, both criteria are summarized in a weighted sum $w^s E_I^* - (1 - w^s) \eta^*$ with a preselected weight $0 \leq w^s \leq 1$ kept constant during optimization. This results in the global optimization problem

$$\min_{\mathbf{s}^w \in S} f \quad \text{s.t.} \quad S = \{\mathbf{s}^w \in \mathbb{R}^8 \mid \mathbf{s}^{w,l} \leq \mathbf{s}^w \leq \mathbf{s}^{w,u}\} \quad (18)$$

with

$$f = w^s E_I^* - (1 - w^s) \eta^* \quad (19)$$

Fig. 3 Optimization scheme for application example



for the global variable vector

$$s^w = [\tilde{y}_{C|B}^T, p_s^T, w]^T \tag{20}$$

summarizing goals $\tilde{y}_{C|B}$ for the coupling (14), shared design variables (17) and the weight w prescribed for the combustor subprocess. The coupling towards the turbine $\tilde{y}_{B|T}$ is kept constant.

According to Eq. (8) and Figs. 2 and 3, the compressor optimization subproblem for the application test case is defined as a single objective design problem

$$\min_{p_c \in P_C} (-\eta_c) \text{ s.t. } P_C = \{p_c \in \mathbb{R}^{57} \mid h_c(p_c) \leq 0, p_c^l \leq p_c \leq p_c^u, \left\| \begin{bmatrix} p_{30}(p_c) \\ T_{30}(p_c) \\ M_{30}(p_c) \\ \alpha_{ex}(p_c) \end{bmatrix} - \tilde{y}_{C|B} \right\| \leq \varepsilon_C, \|n_{HP}(p_c) - \tilde{y}_{C|T}\| \leq \varepsilon_C\} . \tag{21}$$

The coupling vector $\tilde{y}_{C|B}$ as prescribed by the global optimizer enforces the downstream coupling towards the combustor. The coupling from the turbine is realized by directly providing the prescribed turbine power P_T as $\tilde{y}_{T|C}$ to the compressor, where this constraint would be usually enforced in the turbine optimization process. Mechanical losses are not considered at this stage. The coupling of compressor and turbine $\tilde{y}_{C|T}$ represented by high pressure shaft speed n_{HP} is also enforced as a constraint.

The compressor design process used here is based on a component design process developed by Pöhlmann (2015). It uses a fully parameterized compressor gas path which is analysed by a throughflow solver. The design is then optimized using a CMA-ES (Covariance Matrix Adaptation - Evolution Strategy), optimizer

Hansen (2006), without surrogates with regards to local objectives and boundary conditions. The design variables \mathbf{p}_C involve quantities determining the annulus shape and flow behaviour.

The current wall clock time for a compressor subsystem optimization is about six hours. This is due to the complex nature of the subsystem optimization problem, where about 6000 iterations are required for a converged solution. The runtime of this subsystem optimization process, therefore, drives the global computation time according to the number of global iterations. This gives a strong incentive to improve the runtime of all subprocesses as well as to reduce the number of iterations required by the global optimizer.

The subsystem optimization problem for the combustor subprocess is derived from Eq. (9) as

$$\min_{\mathbf{p}_B \in P_B} (wE_I - (1-w)\eta_B) \text{ s.t. } P_B = \left\{ \mathbf{p}_B \in \mathbb{R}^6 \mid \mathbf{h}_B(\mathbf{p}_B) \leq \mathbf{0}, \mathbf{p}_B^l \leq \mathbf{p}_B \leq \mathbf{p}_B^u, \left\| \begin{bmatrix} p_{40}(\mathbf{p}_B) \\ T_{i40}(\mathbf{p}_B) \\ M_{40}(\mathbf{p}_B) \end{bmatrix} - \tilde{\mathbf{y}}_{B|T} \right\| \leq \varepsilon_B \right\} \quad (22)$$

where the coupling towards the turbine needs to meet a value $\tilde{\mathbf{y}}_{B|T}$. The weight w is prescribed by the global optimizer and is constant during the local optimization. The combustor optimization subprocess is based on Angersbach and Bestle (2014) providing tools for preliminary combustor design. Here the local optimization process optimizes the airflow distribution throughout the combustion chamber to improve efficiency and emissions as well as to meet local constraints like the pressure loss distribution. The process is modified to work in the environment of a global optimization process and uses a CMA-ES algorithm to solve the local optimization problem with surrogates. The number of iterations required is lower compared to the compressor subsystem problem, but each individual iteration takes more computation time due to file system operations dictated by the use of legacy code.

The global problem, as described in Eq. (18), is solved with the local-Multistart Stochastic Response Surface (MSRS) algorithm by Regis and Shoemaker (2007). It fulfils the demand for a small number of system evaluations, which here are challenging and time-consuming subsystem optimizations, through the application of a response surface based search. The algorithm starts with an initial Design of Experiments (DoE) with N_S points to be evaluated, which are equally distributed in the global design space \mathbf{s}^w by Latin hypercube sampling (LHS). The ranges of the global variables are defined through a reference configuration. The subsystem optimization is carried out at these points and the resulting objective function values η_C^* , η_B^* , E_I^* are approximated with radial basis functions (RBF), Bouhamidi and Méhauté (2004). A parallel calculation is possible during the DoE and allows for a reduced overall computation time. From the calculated set of expensive function evaluations the point \mathbf{s}^{w*} with the best objective function value is chosen. A set of candidate points $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_n\}$ is created around the chosen point as random perturbations with zero mean and covariance matrix $\sigma^2 I_8$ with σ^2 being the search radius. The

points are then evaluated on the response surface to determine the most promising candidate point. This is done by assessing the predicted function value $\hat{f}(\mathbf{c}_i)$ of each candidate point and also by calculating the minimal distance $d(\mathbf{c}_i)$ to each point of the DoE. To find the most promising candidate point, a weighted score is calculated as

$$W(\mathbf{c}_i) = w_f g_f(\mathbf{c}_i) + w_d g_d(\mathbf{c}_i) \quad (23)$$

involving the normalized objective criterion

$$g_f(\mathbf{c}_i) = \frac{\hat{f}(\mathbf{c}_i) - f_{min}}{f_{max} - f_{min}} \quad (24)$$

and the distance criterion

$$g_d(\mathbf{c}_i) = \frac{d_{max} - d(\mathbf{c}_i)}{d_{max} - d_{min}} \quad (25)$$

weighted by w_f and w_d , respectively, where $w_f + w_d = 1$. The distance criterion guarantees sufficient distance to previously calculated points. A new point is selected for an expensive calculation $\mathbf{c}_i \in \mathcal{C}$ through the weighted score W , and the response surface is updated accordingly. The counter of non-improved steps n_e is raised if \mathbf{c}_i is not feasible or no improvement is found, i.e., $f(\mathbf{c}^*) \geq f(\mathbf{s}^{v*})$. To improve convergence, the search radius is reduced by a factor of 0.5 if n_e reaches a predefined value N_e . This is repeated until a minimum radius is reached where the search is restarted with a new DoE or terminated.

Due to the high complexity of the problem, the required number of points in the DoE is hard to assess beforehand. To alleviate this problem, an augmented Latin Hypercube Sampling algorithm (aLHS) based on Dalbey and Karystinos (2010) is integrated into the calculation procedure. It allows for an extension of the original sampling whilst still keeping results of the expensive calculations already performed and preserves the characteristics of a LHS. To determine the need for an augmentation of the original sampling, the response surface fitted to the results of the original DoE is evaluated through a leave-one-out cross-validation and the resulting Spearman correlation coefficient is compared to a predefined goal value. If the correlation is not sufficient, the number of points in the DoE as well as the number of iterations is augmented by a predefined factor and tested again with the added points.

Result Discussion

The test case optimization is based on a reference engine configuration taken from a representative industry application. The presented result is obtained after 50 global iterations which is reached after 120h showing the importance of an efficient global optimization algorithm due to the long run times. Out of the 50 calculations, 45 converged for both compressor and combustor, yielding valid results to be used in

the global optimization. This equals to a convergence rate of 90%. A subsystem optimization is considered to be converged if all constraints are fulfilled and there is no further improvement with respect to the local objectives. In all the cases of non-convergence, only one subprocess per iteration did not converge, mostly due to restrictive geometric variable inputs resulting in a failure to reach given boundary conditions. Non-converged points are not used for updating the response surface of the MSRS, but as part of the distance criterion to prevent a recalculation of narrow design points. The results were obtained using the settings given in Table 1. The used local-MSRS algorithm does not change the weights for distance and fitness over the course of the optimization, thus forcing a smaller clustering radius due to the low impact of the distance weight. Further the small search radius improves local sampling density around the best point of the DoE. The number of allowed non-improved results is set to 8 to provide sufficient exploration of the high dimensional variable space without wasting too many calculations before the search radius is halved. The normalized fitness is defined as $f_{norm} = f/f_{ref}$ with f being the fitness of the MSRS algorithm according to Eq. (9).

Figure 4 shows the current fitness value f_{norm} of the global configuration for the duration of optimization, where the iterations performed during the initial DoE are

Table 1 Global optimization settings

Number of global iterations n_{glob}	50
Number of points in the initial DoE N_s	41
Initial search radius σ^2	0.025
Distance weight w_f	0.95
Fitness weight w_d	0.05
Maximum number of non-improved results N_e	8

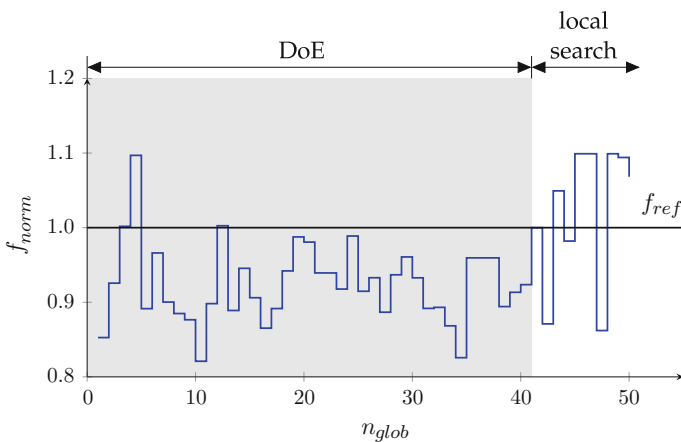


Fig. 4 Current fitness value for the global optimization, compared to the reference fitness f_{ref}

marked in grey. The fitness is optimized compared to the reference configuration with fitness f_{ref} shown by line $f_{norm} = \lambda$. The highest positive value of f_{norm} represents the best configuration due to the normalization. A first improvement regarding overall performance as defined by the objectives in Eqs. (1) and (2) and represented by the fitness value is found during the DoE, which is then further improved by the global optimizer via localized search. The optimizer uses information from the DoE to identify the approximate location of the global optimum. After that, the local clustering around the presumed best point improves the quality of the response surface approximation locally, enforced through the initial search radius σ^2 and the distance weight w_f . This enables the algorithm to find a true global optimum close to the presumed point, but it limits the amount of improvement to be achieved after the end of the DoE, depending on the amount of improvement already found during the DoE. Additionally the number of global iterations used in this example represents a lower boundary due to the long calculation time of the component processes. A further improvement is to be expected for further computation. Every subsystem configuration is locally optimal for the given configuration, but may not present the best value overall found during the optimization for the given local problem. The improvement achieved during optimization is considerable regarding the fact that the reference configuration already represents a capable design.

As the local-MSRS algorithm tends to focus on the specific region of highest presumed performance, local variations after the end of the DoE are quite small; see Fig. 5 showing the variations of the aerodynamic coupling variables between compressor and combustor. The design settles on a configuration with relatively low exit pressure allowing for good compressor efficiency whilst not having strong effects on the combustor performance. The compressor exit temperature is prescribed at relatively high values, which poses a less severe constraint on the compressor compared to a low exit temperature. The Mach number is forced to low compressor

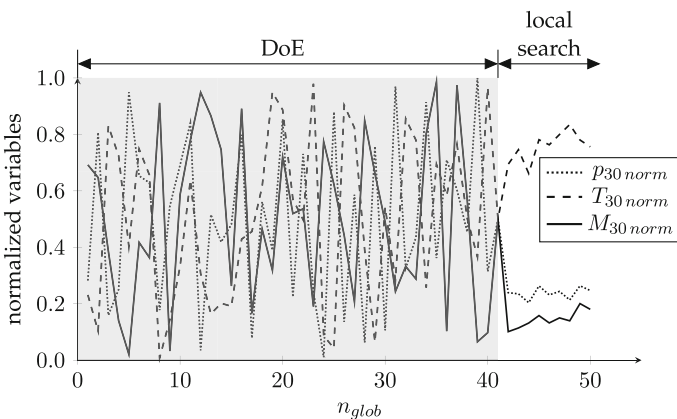


Fig. 5 Normalized compressor exit and combustor inlet quantities as prescribed by the global optimizer

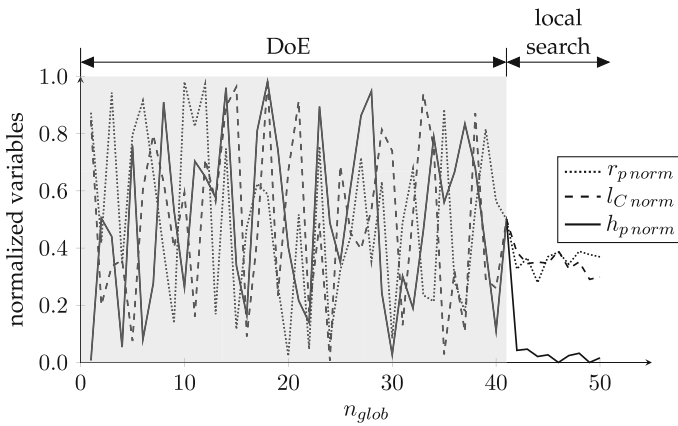


Fig. 6 Normalized geometric quantities as prescribed by the global optimizer

exit values posing a challenge on the compressor optimization subprocess where good combustor performance is associated with low inflow Mach numbers.

The distribution of the geometric coupling variables is shown in Fig. 6, with a fast focus after the DoE. The geometric configuration promising the highest performance is characterized by a low compressor gas path exit height. This mainly influences the Mach number at the compressor exit via the overall compressor flow area. The medium radial height of the compressor gas path is permissible due to the fact that requirements for compressor exit pressure are relatively low. The length requirements for the compressor are kept at a medium level as well, in turn influenced by the required pressure at the compressor-combustor interface.

Conclusions and Outlook

The process presented in this paper enables the solution of a complex overall optimization problem through decoupling whilst using legacy code, based on the current state of the art of preliminary engine design. Until now, this is done with fixed interfaces, which limit the overall design space. The proposed method removes this limitation and uses flexible interfaces to explore new configurations. The performance goal is defined as the overall core engine efficiency as a product of the component efficiencies. The problem is represented by a simplified model of a industrial application case, for which an optimized configuration is found. The use of an efficient algorithm guarantees a minimum number of overall iterations. The fact that the subprocesses are not simple system evaluations but rather complex optimizations themselves, and thus retain an uncertainty regarding their results, poses a special challenge to the overall optimization loop. Furthermore, a substitution of the current weighted sum approach in the global optimization by a real multiobjective global

optimizer is planned. Finally a step to implement the multiobjective coupling as presented by Lockan et al. (2017) is planned to improve the performance of the subprocess interfaces.

Acknowledgements This work has been carried out in collaboration with Rolls-Royce Deutschland as part of the research project VITIV (Virtual Turbomachinery, Proj.-No. 80164702) funded by the State of Brandenburg, the European Regional Development Fund, and Rolls-Royce Deutschland. Rolls-Royce Deutschlands permission to publish this work is greatly acknowledged.

References

- Adelman HM, Haftka RT (1986) Sensitivity analysis of discrete structural systems. *AIAA J* 24(5):823–832
- Angersbach A, Bestle D (2014) Optimization of air distribution in a preliminary design stage of an aero-engine combustor. In: 14th AIAA aviation technology, integration, and operations conference, AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics
- Bouhamidi A, Méhauté AL (2004) Radial basis functions under tension. *J Approximation Theory* 127(2):135–154
- Dalbey K, Karystinos G (2010) Fast generation of space-filling Latin hypercube sample designs. In: 13th AIAA/ISSMO multidisciplinary analysis optimization conference, vol 2010-9085
- Hansen N (2006) The CMA evolution strategy: a comparing review. Springer, Berlin, pp 75–102
- ICAO Airport Air Quality Manual (2011) <http://www.icao.int/environmental-protection/Documents/Publications/FINAL.Doc%209889.Corrigendum.en.PDF>. Accessed 15 Mar 2017
- Lockan M, Bestle D, Janke C, Meyer M (2017) Optimization of coupled system components using approximations of interface quantities. In: Proceedings of the ASME Turbo Expo 2017: turbine technical conference and exposition, Charlotte, pp. GT 2017–64,135
- Pöhlmann F (2015) Optimization and coupling strategies for codes of different fidelity to automate an aerodynamic compressor design process. PhD thesis, Brandenburg University of Technology, Shaker, Aachen
- Regis RG, Shoemaker CA (2007) A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J Comput* 19(4):497–509
- Sellar R, Batill S, Renaud J (1996) Response surface based, concurrent subspace optimization for multidisciplinary system design. In: 34th aerospace sciences meeting and exhibit, p 714
- Sobieszcanski-Sobieski J (1989) Optimization by decomposition: a step from hierarchic to non-hierarchic systems. NASA STI/Recon Technical Report N 89
- Sobieszcanski-Sobieski J, Agte JS, Sandusky R (1998) Bi-level integrated system synthesis. NASA Technical Memorandum 1998-208715
- Sobieszcanski-Sobieski J, Altus TD, Phillips M, Sandusky R (2003) Bilevel integrated system synthesis for concurrent and distributed processing. *AIAA J* 41(10):1996–2003

Progresses in Fluid-Structure Interaction and Structural Optimization Numerical Tools Within the EU CS RIBES Project



Marco Evangelos Biancolini, Ubaldo Cella, Corrado Groth, Andrea Chiappa, Francesco Giorgetti and Fabrizio Nicolosi

Abstract The capability to reduce the structural weight of aircrafts, and consequently the fuel consumption, is related to the accuracy of numerical tools and to the efficiency of design methodologies available. In particular, the capability to model the interaction of the several mechanisms involved in physics phenomena represents a key point in the development of engineering design tools. Typical examples are FSI (Fluid-Structure Interaction) analyses in which the capability to properly capture the behaviour of aeroelastic phenomena is crucial. Furthermore, the enhancement of environments able to include structural shape optimizations represents a significant step forward in the development of greener aircrafts. The objectives of the EU RIBES (Radial basis functions at fluid Interface Boundaries to Envelope flow results for advanced Structural analysis) project was to reduce the uncertainty in CFD (Computational Fluid Dynamics)-CSM (Computational Structural Mechanics) aeroelastic analysis numerical methodologies, enhancing the coupling between fluid-dynamic and structural solvers, to improve the confidence on their accuracy and to progress in the development of structural optimization tools. At this aim, the project was focused on the development of an accurate load mapping procedure, on the implementation of an innovative workflow for structural shape optimization and on experimental validation of FSI (Fluid-Structure Interaction) methodologies. Radial Basis Functions (RBF) supply the mathematical foundation for the first two topics.

M. E. Biancolini · U. Cella · C. Groth · A. Chiappa (✉) · F. Giorgetti
University of Rome “Tor Vergata”, Rome, Italy
e-mail: andrea.chiappa@uniroma2.it

U. Cella
Design Methods, Messina, Italy

F. Nicolosi
University of Naples “Federico II”, Naples, Italy

This paper summarizes the results achieved by the project, describes the developed optimization tool and details the experimental campaign conducted to generate a database of measurements on a typical realistic aeronautical wing structure.

Introduction

The “RIBES”¹ (Radial basis functions at fluid Interface Boundaries to Envelope flow results for advanced Structural analysis) project started in December 2014 and was completed December 2016. Its objective was the enhancement of FSI (Fluid Structure Interaction) analysis methodologies by the development and the validation of an accurate coupling between structural and fluid dynamic tools and the development of a structural optimization software capable to modify both thickness properties and shape. The project, led by the University of Rome “Tor Vergata”, progressed thorough the three main topics:

1. the development of an accurate and efficient load mapping procedure to transfer vector fields from CFD (Computational Fluid Mechanics) to FEM (Finite Element Method) solvers;
2. the implementation of a structural shape optimization tool;
3. the setup of an aeroelastic experimental campaign for FSI methodologies validation.

An innovative aspect of the methods developed is the adoption of Radial Basis Functions (RBF) as a mathematical framework with which to face the problem of post-processing of multivariate data and solution interpolation between non-matching domains.

Partners of research were the *RBF Morph*² software vendor, whose technology is at the base of the implemented methodologies, the aerospace consulting engineering firm *Design Methods*³ which supported the experimental activities and the FSI methods validation and the University of Naples “Federico II” that was in charge of the experimental measurements.

The remainder of the paper is organized as follows: Section “[Radial Basis Functions](#)” supplies a brief background about RBF, Section “[Load Mapping Procedure](#)” describes the load mapping procedure and its application to an aeronautical test bench (in Section “[Mapping Procedure Validation](#)”). The optimization strategy is outlined in Section “[Structural Optimization Environment](#)”

¹<http://ribes-project.eu/project>.

²www.rbf-morph.com.

³www.designmethods.aero.

along with a practical example presented in Section “[Testing of the Optimization Procedure](#)”. Section “[Experimental Campaign](#)” is dedicated to the experimental campaign. Finally, conclusions are drawn in Section “[Conclusions](#)”. A few pictures referring to the experimental and numerical activities are reported in the Appendix.

Radial Basis Functions

Radial Basis Functions are powerful mathematical functions able to interpolate, giving the exact values in the original points, functions defined at discrete points only (source points) (Buhmann 2004). The interpolation quality and its behaviour depends on the chosen RBF.

A linear system (of order equal to the number of source points introduced) needs to be solved for coefficients calculation (De Boer et al. 2007). Once the unknown coefficients are calculated, the field value at an arbitrary point inside or outside the domain (interpolation/extrapolation) is expressed as the summation of the radial contribution of each source point (if the point falls inside the influence domain). An interpolation function composed by a radial basis and a polynomial is defined as follows:

$$s(\mathbf{x}) = \sum_{i=1}^N \gamma_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + h(\mathbf{x}) \tag{1}$$

The minimal degree of polynomial h depends on the choice of the basis function. A unique interpolant exists (Buhmann 2004) if the basis function is a conditionally positive definite function. If the basis functions are conditionally positive definite of order $m = 2$, a linear polynomial can be used in a three-dimensional space:

$$h(\mathbf{x}) = \beta + \beta_1 x + \beta_2 y + \beta_3 z \tag{2}$$

In order to solve the whole system with both contributions (1) and (2), an additional condition is required. This last comes from orthogonality between the weights of the radial basis and those of the polynomial supplement. The values for the coefficients γ of RBF and the coefficients β of the linear polynomial can be obtained by solving the system

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta} \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \tag{3}$$

where \mathbf{g} is the vector of known values at the source points. \mathbf{M} is the interpolation matrix defined calculating all the radial interactions between source points

$$M_{ij} = \phi(\|\mathbf{x}_{k_i} - \mathbf{x}_{k_j}\|) \quad 1 \leq i \quad j \leq N \tag{4}$$

and \mathbf{P} is the constraint matrix

$$\mathbf{P} = \begin{pmatrix} 1 & x_{k_1}^0 & y_{k_1}^0 & z_{k_1}^0 \\ 1 & x_{k_2}^0 & y_{k_2}^0 & z_{k_2}^0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{k_N}^0 & y_{k_N}^0 & z_{k_N}^0 \end{pmatrix} \tag{5}$$

RBF have several advantages that makes them very attractive in many areas of science and engineering: surface reconstruction (Casciola et al. 2006), solution of partial differential equations (Fasshauer 1999), image analysis of deformations (Biancolini and Salvini 2012) and mesh morphing, to cite the most popular applications. This last technique (Biancolini 2012) refers to the capability to rearrange an existing mesh into a new shape, leaving the topology unchanged.

Load Mapping Procedure

A critical aspect in CFD-CSM based aeroelastic analysis is represented by the technique with which to transfer the aerodynamic loads from the wet surfaces of the fluid dynamic mesh to the structural domain which, in general, have non-matching discretization on the common boundaries (Jiao and Heath 2004). The forces computed by the CFD analyses are, in fact, extracted from the cells of the CFD walls boundaries in the form of vectors positioned on a cloud of points that will typically differ from the FEM grid points (Fig. 1) on which the loads have to be applied (the grids requirements are, in general, different for FEM and CFD analyses). An interpolation between the two domains is then required with a consequential introduction of an

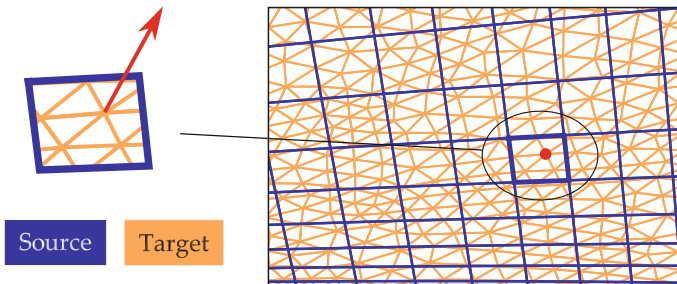


Fig. 1 Example of non-matching meshes

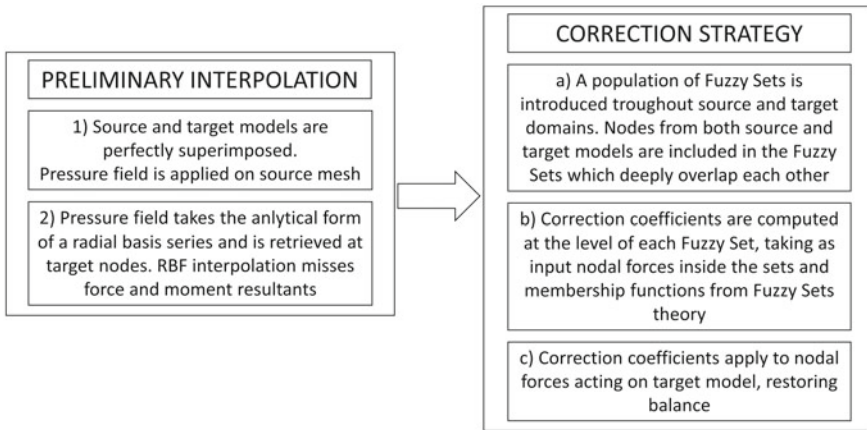


Fig. 2 Main steps of the transfer procedure devised for the RIBES project

error. The minimization of the uncertainty associated to this process is the major concern when defining a mathematical approach addressed to the interpolation task.

A major drawback of RBF is that, in order to obtain the interpolating series, a linear system whose dimensions depend on the number of considered points, needs to be solved. This issue can be conveniently handled by resorting to a proper domain decomposition technique, such as the Partition of Unity (POU) strategy (Babuska and Melenk 1997). The POU method is used to organize the source and target point sets into overlapping subdomains in which the interpolation problem is locally solved. In order to recover the continuity of the field and to guarantee the smoothness of the global solution, the resulting set of local solutions are combined together by a blending function.

Although RBFs proved to be very accurate for interpolation, they cannot ensure load conservation in the case of pressure mapping. This last requirement is satisfied with the introduction of a Fuzzy Sets population (Zadeh 1965) throughout source and target domains, to correct the outcome of RBF interpolation in order to meet local and global balance.

The graph reported in Fig. 2 summarizes the main steps of the proposed transfer procedure.

In Fig. 3 an example of interpolation, in which the pressure is mapped between non-matching nodes mesh using RBF and Fuzzy Sets, is reported. The upper figure reports the source CFD mesh while the target FEM mesh is reported in the lower one.

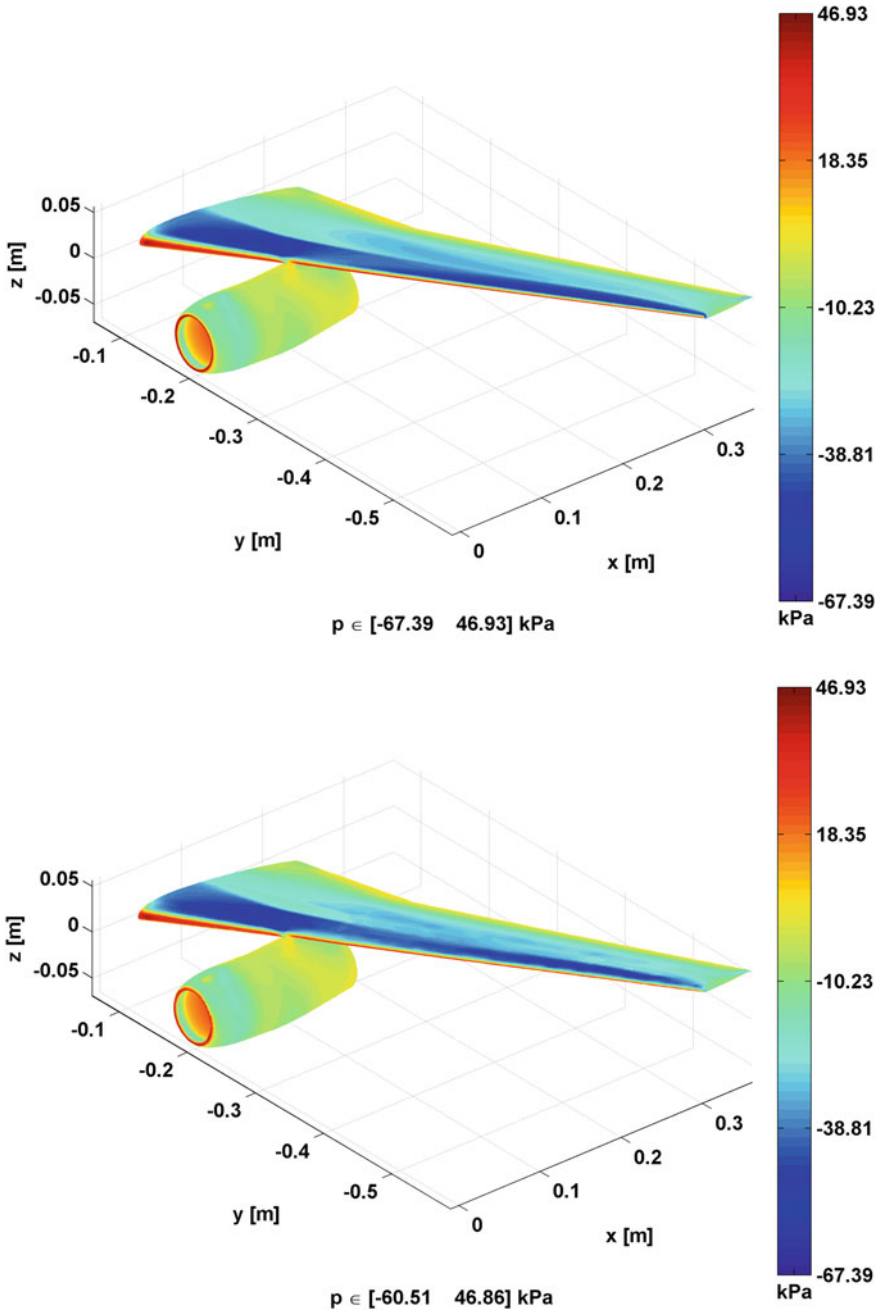


Fig. 3 Example of load mapping interpolation between non-matching domains, source (above) and target (below)

Table 1 Non-corrected interpolation errors between HiReNASD model grids

F_x err (%)	F_y err (%)	F_z err (%)	M_x err (%)	M_y err (%)	M_z err (%)
47.3	6.6	1.0	13.1	13.8	27.8

Table 2 Corrected interpolation errors between HiReNASD model grids

M_x err (%)	M_y err (%)	M_z err (%)
0.01	-0.28	-0.06

Mapping Procedure Validation

The developed mapping procedure was validated using the HiReNASD (High Reynolds Number Aero-Structural Dynamics) test case (Ballmann 2008). It consists in a half wing/fuselage wind tunnel model used in a workshop for aeroelastic numerical methods validation. Both CFD and FEM grids are available. The static pressure field was mapped from the CFD wing surface grid to the wing structural FEM mesh. Table 1 reports the errors introduced by the interpolation process without correction between the two non-matching domains. Mismatches on forces (F) and moments (M) are computed component-wise along x , y and z directions (which appear as subscripts).

Table 2 reports the errors on moments obtained applying the correction procedure implemented within the RIBES project (in which errors on forces are zero for definition). The errors are below 0.3% along all directions.

Structural Optimization Environment

Structural optimization is a strategic topic in aircraft design. The weight reduction is directly connected to the performances improvement and to the “green” behaviour of the machine. Several consolidated methods, able to optimize the thickness of the structural elements, are available on the market. However, the capability to perform an efficient structural shape optimization is still an open issue.

The structural optimization procedure developed within the RIBES project is based on routines that manage the geometric parameters, update the model, perform the FEM analysis and drive the optimization. The possibility to perform both a shape and thickness optimization was implemented by two strategies. Thickness is introduced as variable of design directly taking the control of the *Nastran* bulk data file by the management of the property label in the ASCII file (Fig. 4).

The shape change is applied to the topology directly on the numerical domain by mesh morphing techniques (Fig. 5). Several algorithms have been explored for this task (Sederberg and Parry 1986; Masud et al. 2007). The mesh morphing algorithm

```

$ Femap with NX Nastran Property 1 : long. anteriore
PSHELL 1 6 0.0005 6 0.
$ Femap with NX Nastran Property 2 : long. posteriore
PSHELL 2 6 0.00067 6 0.
$ Femap with NX Nastran Property 3 : centina 1
PSHELL 3 6 0.0005 6 6 0.
$ Femap with NX Nastran Property 4 : centina 2
PSHELL 4 6 0.00051 6 6 0.
$ Femap with NX Nastran Property 5 : centina 3
PSHELL 5 6 0.0005 6 6 0.
    
```

Fig. 4 Bulk data file

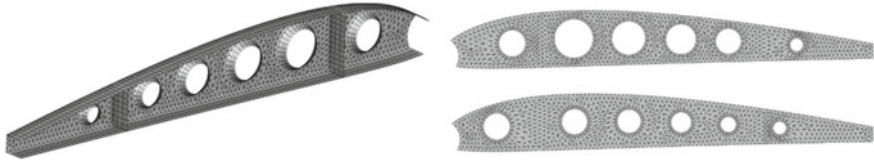


Fig. 5 Example of mesh morphing

implemented in the RIBES software is based on the use of the FEM itself. Shape modifications are applied to an auxiliary model with the same geometry of the original one, the deformed mesh is the result of a FE analysis.

The shape modification is applied to the computational domain as a linear combination of vectors belonging to the FEM model nodal positions space. The parametric formulation of the structural mesh is:

$$\mathbf{X}_{FEM} = \mathbf{X}_{FEM_0} + \eta_s \delta \mathbf{X}_s \tag{6}$$

where \mathbf{X}_{FEM} are the positions of the grid nodes, \mathbf{X}_{FEM_0} are the positions of the nodes of the undeformed mesh and η_s are parameters (that constitute the variables of design) used to amplify the displacement vectors $\delta \mathbf{X}_s$.

The optimum definition is based on filling a DOE (Design Of Experiment) table, with design points selected with the Latin hypercube sampling method (McKay et al. 1979), and on the application of a Response Surface (RS) on which to apply the search algorithm. One of the most efficient approach, for the RS evaluation, is the Kriging method and the RBF interpolation (Jin et al. 2001). The University of Rome “Tor Vergata” developed a RS metamodel based on the RBF using various kernels.

The complete workflow of the structural optimization procedure is sketched in Fig. 6.

The automatic process is included in the dashed frame while the external blocks indicate the setup activities that have to be manually performed in advance. A set of script routines complete the DOE table, drive the computation, extract the solutions, update the FEM setup according to the selected design point, compute the RS and apply the single or multi-objective search algorithm to find the optimum or the Pareto solution. The parameters that is possible to extract to be used to compute the objective functions and/or to implement constraints are the maximum stress (or the maximum

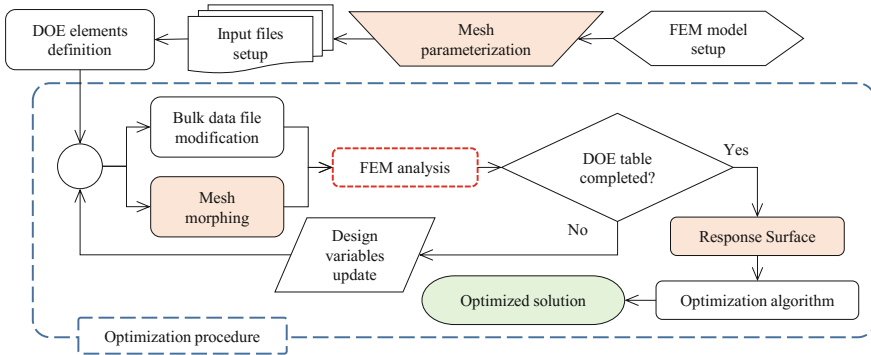


Fig. 6 Workflow of the structural optimization procedure

failure index in the case of composite materials), the maximum displacement (for the entire structure or for each property of the model) and the total mass of the structure.

A Graphical User Interface (GUI) was developed to simplify the problem setup and the postprocessing of solutions. It can be used to automatically generate the input text file expected by the software, evaluate the Pareto front, extract designs and perform other useful operations as plotting the response surface.

Testing of the Optimization Procedure

The procedure described above was tested on the optimization of the RIBES wind tunnel model metal structure (Cella et al. 2015). The test following described was performed on a composite rib obtained taking as a reference the RIBES wing arbitrarily scaled to a realistic full scale.

A preliminary FE analysis was run on a portion of wing box with all its elements (skin, ribs, spars). This structure was constrained at the root section and loaded by forces distributed on a set of points according to the mapping procedure previously described. The displacement field obtained for the boundary of the rib was extracted from the overall model and used to load a new model of the single rib.

Three lightening holes were placed on the rib and reinforcements were extruded around them. Two additional vertical reinforcements were added between the holes. The material of the rib was assumed to be carbon fibre reinforced epoxy pre-preg.

A single objective optimization was setup. The objective function was the minimization of the total mass constraining the structural integrity. The variables of design were:

- Depth of web openings (out of plane) ranged from 0 (flat) to 30 mm;
- Clew depth from 0 (flat) to 30 mm;
- Holes radius from 100 to 140 mm.

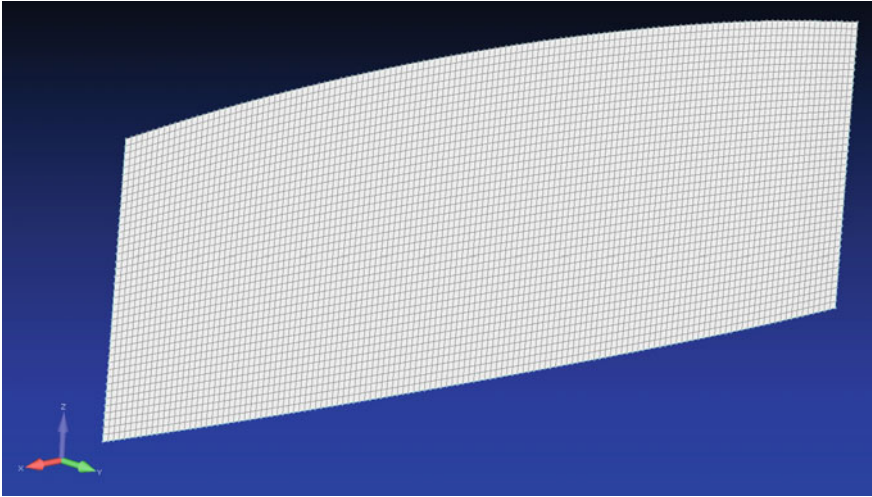


Fig. 7 Baseline rib

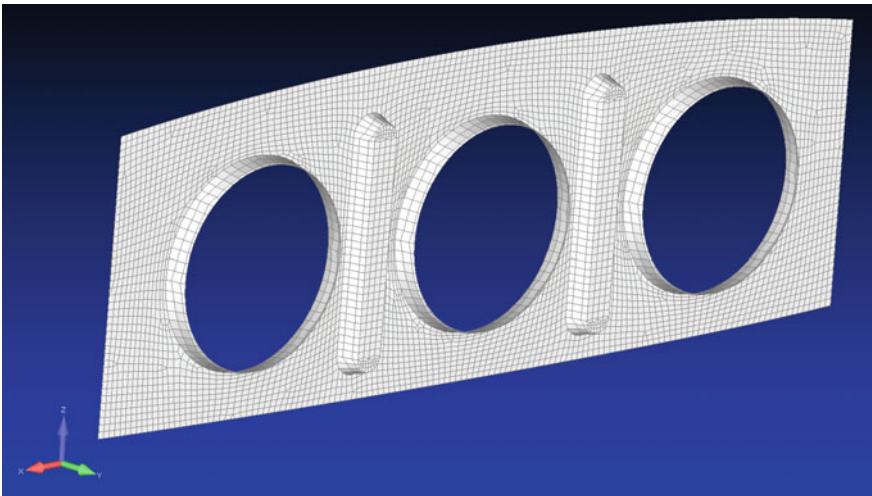


Fig. 8 Optimized rib

A comparison of the rib before and after the application of the proposed modifications is shown in the next two figures (Figs. 7 and 8).

In order to properly populate the design space, 250 structural linear analyses (instability issues were not kept into account) were run. Material strength was evaluated in terms of Failure Index (FI) according to composite materials theory. Results of analyses (in terms of maximum failure indexes and masses) were used as input for the RBF interpolation in order to generate a RS. All points resulted safe from a structural point of view since all maximum failure indexes were lower than 1.

The lowest mass was obtained, as expected, at the maximum value of holes radii. This parameter was the only affecting the solution. The estimated output for this arrangement was $FI=0.52$, $mass=3.35$ kg. As a check, a FEM analysis was run in order to assess the RBF prediction. The obtained results were $FI=0.49$ (percentage error 6%) and $mass=3.36$ kg (percentage error 0.3%).

A deeper verification of the proposed method was performed by evaluating also the configuration with the lowest FI, with a value of 0.40 and a mass equal to 4.08 kg. According to the RBF interpolation, this condition would have been achieved with depth of web openings 30 mm and holes radii of 130 mm. The FEM analysis provided a result ($FI=0.43$, $mass=4.08$ kg) with an error lower than 7%.

It is possible to conclude, for the loading configuration considered, the vertical clews to provide no benefit from a structural point of view. Only the reinforcements around the openings actually gave a contribution to the strength of the rib. It showed, nevertheless, to be not necessary to avoid the risk of failure.

Experimental Campaign

A significant part of the RIBES project resources were allocated on the setup of an extensive aeroelastic wind tunnel test campaign to be performed on a model that replicates a typical metallic aeronautical wing structure. The aim is to generate an experimental base of assessment customized for the validation of the software developed within the project. The task is to produce a database of loads, pressure, stress and deformation measurements that is significant for a realistic aeronautical design problem (<http://ribes-project.eu/experiments/>). This activity was conducted with the collaboration of the University of Naples “Federico II”. The facility adopted was a low speed closed circuit wind tunnel with a test Section “Radial Basis Functions” metres wide and an airflow speed limit of 45 m/s. The model deformation was evaluated by laser scan.

The requirements for the wing tunnel measurements and for the model were:

- *for the tests*
 - static pressure measurements under steady flow conditions;
 - forces and moments measurements;
 - model deformation measurements;
 - stresses measurements on most significant structure locations;
- *for the model*
 - scaled physical model of a metallic wing type of structure;
 - rectangular shape and small thickness;
 - installation of strain gauges and pressure pick-up points at its surface.

The design of a *flexible* model was particular critical because, for manufacturing reasons, the simple scaling of a real wing box to dimensions compatible to a low

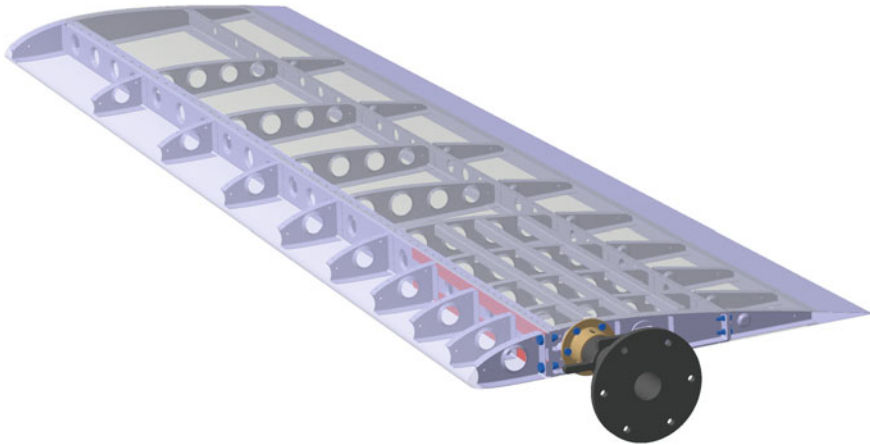


Fig. 9 Detail of the internal structure of the RIBES WT model

speed WT test chamber is impracticable also referring to small aircrafts. The results was necessarily a quite stiff structure.

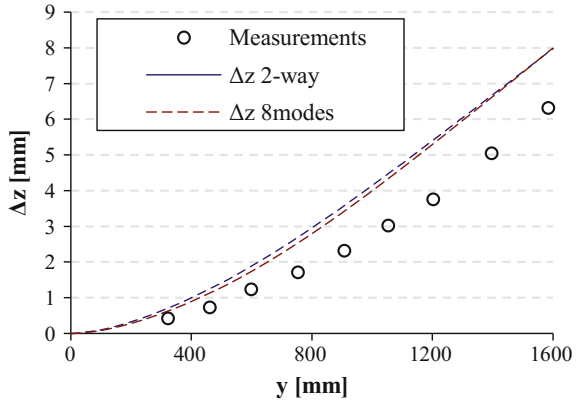
The model consisted in an aluminium alloy tapered wing, with two C shaped spars, ten ribs, 3 upper surface stringers (to prevent buckling) all covered by a skin 0.6 mm thick (Fig. 9). The model has been built following typical aeronautical technique and finally assembled after the installation of both pressure taps and strain gauges. The wing is fixed on the side wall of the test section as a cantilever (Fig. 12 in appendix).

Two FSI analysis methodologies were validated against the RIBES test case: a traditional 2-way (described in Cella and Biancolini 2012), in which the mapping of loads was performed by the procedure developed within the project, and a method based on the modal approach in which the modal base is generated by a set of RBF solutions replicating a set of natural modal shapes of the structure (up to 8 modes were adopted) (Biancolini et al. 2016). The two methods estimated a very similar deformation but both solutions are overestimated compared to measurements (Fig. 10). A hypothesis, to justify this disagreement, might be related to the panels junctions FEM modelling which were simply connected by points. Such model do not account for friction and interference that are present along the junctions surface leading, probably, to model a structure more flexible than the real one.

The set of installed strain gauges (Fig. 11) provided a map of the stress state of the structure. For the numerical/experimental comparison was used the solution of the FEM analysis resulting from the convergence of the 2-way procedure. Figure 13 in appendix reports the visualizations of the normal stresses, in the Y direction and highlights the locations of some of the strain gauges. Table 3 compares the numerical values of the computation with the measurements.

A reasonable matching between measured and computed stresses was, in general, observed. Significant disagreements are, however, present in locations close to the root region (sensors ID numbers higher than 13) where higher stress gradients are

Fig. 10 Numerical-experimental comparison of RIBES WT model displacement



present. The combination of the information deriving from the computed deformation and the comparison of the stress state would suggest that the distribution of the load between the front spar and the skin was not properly captured. A possible explanation might be related to the modelling of the panels junctions, as stated previously. Furthermore, no junction between reinforcements and spar is modelled. This simplification might be also responsible of the not accurate estimation of the loads transferred between spar and skin. Uncertainty might be also introduced by the junction between the two front spar thickening and the root rib. The elements are, in fact, linked by two linchpins which were not modelled in the numerical configuration. Further details on the RIBES aeroelastic numerical/experimental validation activity can be found in (Cella 2016).

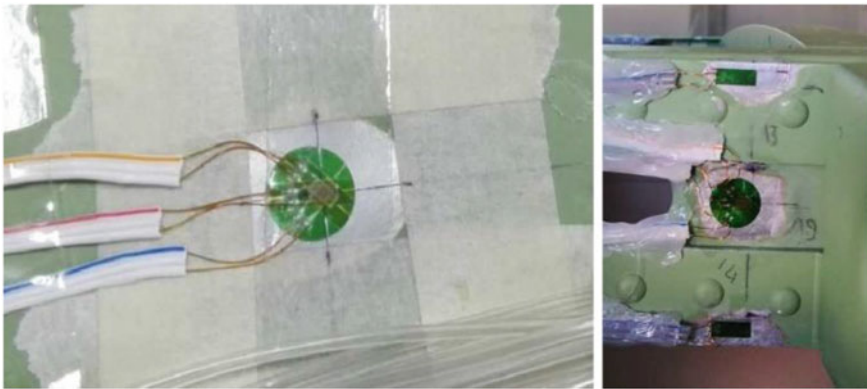


Fig. 11 Details of strain gauges installation of on the WIBES wing structure

Table 3 Measured and computed σ_y at strain gauges location on the RIBES wing

Sensor ID	Position	Measured σ_y (MPa)	FEM σ_y (MPa)
9	Mid front spar up	-12.2	-11
10	Mid front spar low	12.3	-10
11	Mid rear spar up	-8	-8
12	Mid rear spar low	7.5	-6
13	Front spar thickening up	-15.6	-38
14	Front spar thickening low	15	-36
15	Upper skin	-143.2	-21
17	Upper skin	-31.5	-15

Conclusions

The main achievements of the RIBES project concern the developments of an accurate load mapping procedure for CFD-CSM based FSI analyses tools, an optimization software able to face problems of structural shape optimization and a database of aeroelastic measurements (including deformation and stress state) of a typical aeronautical wing structure.

The tools developed within the project offer several advantages. The mapping procedure can be easily automatized and integrated in 2-way FSI workflow to create optimization design environments. The FEM tool capability to parametrize and update the structural properties, combined with the CFD mesh adaptation to the FEM deformation in a closed loop, provides the possibility to create a shape design method that combines aerodynamic and structural optimizations in a single environment. Such a tool can be adopted to face static and dynamic aero-structural designs enabling multi objective/multi physics optimisation in which performances coming from different solutions are pursued steering the same common reference geometry.

All measurements, geometries, numerical models and solutions of numerical/experimental activities performed within the RIBES project are available on line to the scientific community (www.ribes-project.eu). The wish is the RIBES test case to constitute an enhancement for information sharing between scientists and a framework for further discussions, research activities, proposals and collaborations.

Acknowledgements The RIBES project was funded by the European Union within the 7th Framework aeronautics programme JTI-CS-GRA (Joint Technology Initiatives—Clean Sky—Green Regional Aircraft) under Grant Agreement no. 632556.

Appendix

See Figs. 11, 12 and 13.

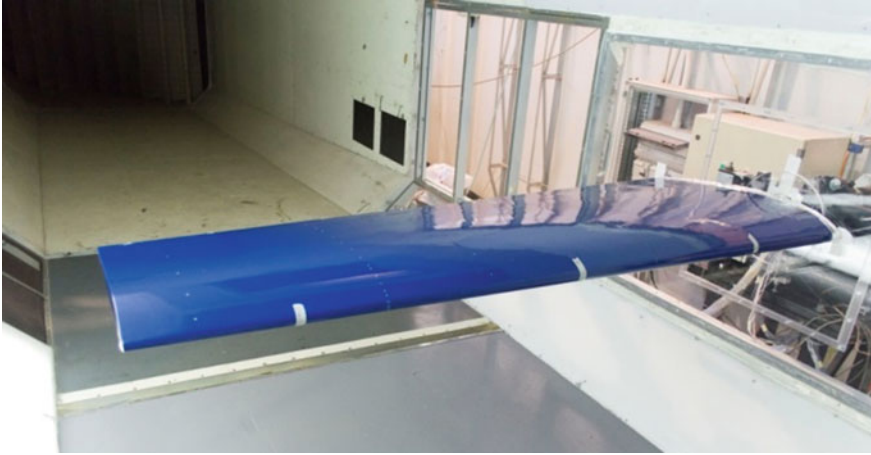


Fig. 12 RIBES WT model installed in the test section

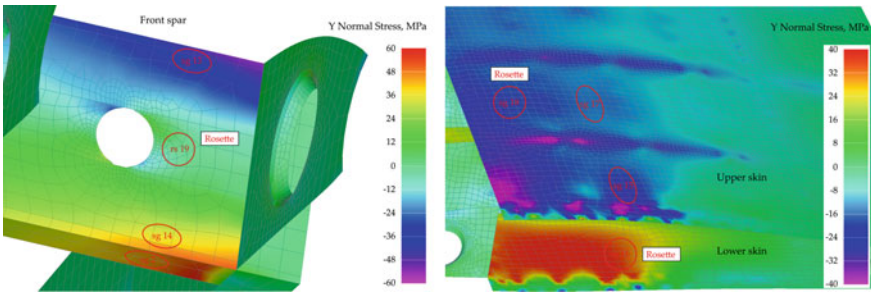


Fig. 13 FEM solution of the 2-way FSI analysis on the RIBES WT model

References

- Babuska I-M, Melenk J-M (1997) The partition of unity method. *Int J Numer Methods Eng* 40(4):727–758
- Ballmann J (2008) Experimental analysis of high Reynolds number structural dynamics in ETW. In: 46th AIAA aerospace sciences meeting and exhibit, number AIAA 2008-841, Reno, Nevada (US), 7–10 Jan 2008
- Biancolini ME (2012) Mesh morphing and smoothing by means of radial basis functions (RBF): a practical example using fluent and RBF Morph. *Handbook of research on computational science and engineering: theory and practice*, IGI Global
- Biancolini ME, Salvini P (2012) Radial basis functions for the image analysis of deformations. In: *Computational modelling of objects represented in images: fundamentals, methods and applications III proceedings of the international symposium*. CRC Press, Boca Raton, FL, pp 361–365
- Biancolini ME, Cella U, Groth C, Genta M (2016) Static aeroelastic analysis of an aircraft wind-tunnel model by means of modal RBF mesh updating. *ASCE's J Aerosp Eng* 29(6)
- Buhmann M-D (2004) *Radial basis functions: theory and implementations*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, Cambridge
- Casciola G, Lazzaro D, Montefusco LB, Morigi S (2006) Shape preserving surface reconstruction using locally anisotropic radial basis function interpolants. *Comput Math Appl* 51(8):1185–1198. <https://doi.org/10.1016/j.camwa.2006.04.002>
- Cella U Setup and validation of high fidelity aeroelastic analysis methods based on RBF mesh morphing. PhD thesis, University of Rome “Tor Vergata”, cycle XXIX, AA 2015/16
- Cella U, Biancolini ME (2012) Aeroelastic analysis of aircraft wind-tunnel model coupling structural and fluid dynamic codes. *AIAA J Aircr* 49(2):407–414
- Cella U, Biancolini ME, Groth C, Chiappa A, Beltramme D (2015) Development and validation of numerical tools for FSI analysis and structural optimization: the EU RIBES project status. In: *AIAS 44th National Congress*, 2–5 Sept 2015, Messina, Italy
- De Boer A, van der Schoot MS, Bijl H (2007) Mesh deformation based on radial basis function interpolation. *Comput Struct* 85(11–14):784–795
- Fasshauer GE (1999) Solving differential equations with radial basis functions: multilevel methods and smoothing. *Adv Comput Math* 11:139–159
- Jiao X, Heath M-T (2004) Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *Int J Numer Meth Eng* 61(14):2402–2427
- Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct Multi Optim* 23(1):1–13
- Masud A, Bhanabagvanwala M, Khurram RA (2007) An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction. *Comput Fluids* 36(1):77–91
- McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Sederberg TW, Parry SR (1986) Free-form deformation of solid geometric models. In: Evans DC and Athay RJ (eds) *13th conference on computer graphics and interactive techniques*. New York, pp 151–160
- Zadeh L-A (1965) Fuzzy sets. *Inf Control* 8(3):338–353