

NOTES ON NUMERICAL FLUID
MECHANICS AND MULTIDISCIPLINARY
DESIGN · VOLUME 122

Management and Minimisation of Uncertainties and Errors in Numerical Aerodynamics

Results of the German Collaborative
Project MUNA

Bernhard Eisfeld · Holger Barnewitz
Willy Fritz · Frank Thiele (Eds.)

 Springer

Series Editors

Prof. Dr. Wolfgang Schröder
(General Editor), RWTH Aachen, Lehrstuhl für Strömungslehre und Aerodynamisches
Institut, Wüllnerstr. 5a, 52062 Aachen, Germany
E-mail: office@aia.rwth-aachen.de

Prof. Dr. Ir. Bendiks Jan Boersma
Chair of Energytechnology, Delft University of Technology, Leeghwaterstraat 44,
2628 CA Delft, The Netherlands
E-mail: b.j.boersma@tudelft.nl

Prof. Dr. Kozo Fujii
Space Transportation Research Division, The Institute of Space and Astronautical Science,
3-1-1, Yoshinodai, Sagamihara, Kanagawa, 229-8510, Japan
E-mail: fujii@flab.eng.isas.jaxa.jp

Dr. Werner Haase
Höhenkirchener Str. 19d, D-85662 Hohenbrunn, Germany
E-mail: whac@haa.se

Prof. Dr. Michael A. Leschziner
Aeronautics Department, Imperial College of Science Technology and Medicine,
Prince Consort Road, London SW7 2BY, UK
E-mail: mike.leschziner@ic.ac.uk

Prof. Dr. Jacques Periaux
38, Boulevard de Reuilly, F-75012 Paris, France
E-mail: jperiaux@free.fr

Prof. Dr. Sergio Pirozzoli
Dipartimento di Meccanica e Aeronautica, Università di Roma "La Sapienza",
Via Eudossiana 18, 00184, Roma, Italy
E-mail: sergio.pirozzoli@uniroma1.it

Prof. Dr. Arthur Rizzi
Department of Aeronautics, KTH Royal Institute of Technology, Teknikringen 8,
S-10044 Stockholm, Sweden
E-mail: rizzi@aero.kth.se

Dr. Bernard Roux
L3M - IMT La Jetée, Technopole de Chateau-Gombert, F-13451 Marseille Cedex 20, France
E-mail: broux@l3m.univ-mrs.fr

Prof. Dr. Yuri I. Shokin
Institute of Computational Technologies, Siberian Branch of the Russian Academy
of Sciences, Ac. Lavrentyeva Ave. 6, 630090 Novosibirsk, Russia
E-mail: shokin@ict.nsc.ru

For further volumes:
<http://www.springer.com/series/4629>

Management and Minimisation of Uncertainties and Errors in Numerical Aerodynamics

Results of the German Collaborative
Project MUNA

Bernhard Eisfeld, Holger Barnewitz,
Willy Fritz, and Frank Thiele (Eds.)

 Springer

Editors

Dr. Bernhard Eisfeld
Institute of Aerodynamics and
Flow Technology
German Aerospace Center (DLR)
Braunschweig
Germany

Dipl.-Phys. Holger Barnewitz
Airbus Operations GmbH
Bremen
Germany

Willy Fritz
EADS Deutschland GmbH Cassidian
Manching
Germany

Prof. Dr. Frank Thiele
Institut für Strömungsmechanik
und Technische Akustik
Berlin
Germany

ISSN 1612-2909

ISBN 978-3-642-36184-5

DOI 10.1007/978-3-642-36185-2

Springer Heidelberg New York Dordrecht London

e-ISSN 1860-0824

e-ISBN 978-3-642-36185-2

Library of Congress Control Number: 2012956291

© Springer-Verlag Berlin Heidelberg 2013

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. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

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.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Continuous development of numerical methods over the last decades together with increasing available computer power has made Computational Fluid Dynamics (CFD) a key technology in modern aircraft development. The results obtained over the last decades give rise to the vision that future aircraft design may almost entirely rely on numerical simulations. This shift in paradigm will not only dramatically change the engineering process in itself, but also the requirements on the simulation data. In particular, safe estimates of the errors and uncertainties of the simulation results will have to be provided, similar to error bars in experimental data from the windtunnel.

In order to meet such future needs, the collaborative research project MUNA – Management and Minimisation of Errors and Uncertainties in Numerical Aerodynamics – has been initiated within the 4th German Aeronautics Research Program (Luftfahrtforschungsprogramm). Following the predecessor projects MEGAFLOW and MEGADESIGN, altogether 12 partners from industry (Airbus, Casidian, Eurocopter), research organisations (DLR, Institute of Aerodynamics and Flow Technology) and universities (RWTH Aachen: Institute of Aerodynamics and Institute of Computational Analysis of Technical Systems; TU Berlin: Institute of Fluid Mechanics and Technical Acoustics; TU Braunschweig: Institute of Fluid Mechanics, Institute of Aircraft Design and Lightweight Structures and Institute of Scientific Computing; University Stuttgart: Institute of Aerodynamics and Gasdynamics; University Trier: Department of Mathematics) have been developing and applying methods, addressing errors and uncertainties of various kind, typically encountered in CFD simulations. For this purpose, the DLR TAU code was provided as major simulation tool.

In a first step the partners jointly collected possible sources of errors and uncertainties, where the computational grid, turbulence modeling, the numerical accuracy and geometrical issues in the context of coupled multidisciplinary simulations have been identified as important. Consequently, the research activities of the first project phase have been concentrating on the respective areas, which is also reflected by the organisation of the book.

In the second phase of the project, stochastic uncertainties, also called aleatory, have been addressed. Therefore the final part of the book is devoted to various activities towards efficiently computing statistical output quantities due to uncertainties in the input parameters, including methods for the robust design under geometrical uncertainties.

The results of the project have been presented to the public during two Workshops held on March 23th and 24th 2010 and on October 25th 2012. The current book documents the results achieved by the partners.

The editor is indebted to all co-workers of the project, in particular to the members of the steering committee, Holger Barnewitz (Airbus), Willy Fritz (Cassidian) and Prof. Dr. Frank Thiele (TU Berlin), for their contributions and inevitable support in making MUNA a success. Furthermore the editor wants to thank the general editor of the Springer series “Notes on Numerical Fluid Mechanics and Multidisciplinary Design”, Prof. Dr. W. Schröder, and the staff of the Springer-Verlag for the opportunity to publish the technical results of the MUNA project.

Finally the funding of the partner activities by the German Ministry of Economics within the 4th German Aeronautics Research Program is gratefully acknowledged.

Braunschweig
November 2012

Bernhard Eisfeld

Contents

Part I Mesh Generation and Manipulation

Methods and Strategies for the Detection and Management of Grid Induced Uncertainties in Numerical Aerodynamics 3

E. Mazlum, R. Radespiel

1	Identification of Sources of Error Induced by Geometry and Discretisation	3
1.1	Numerical Method	4
1.2	Analysis in 2D	4
1.3	Uncertainties Induced by Variation of the Trailing-Edge Geometry	5
1.4	Uncertainties Induced by the Grid Topology	6
1.5	Effects of the Hybrid Border on the Flow Solution	8
2	Methods for the Detection of Improperly Discretised Grid Regions	12
2.1	Error Indicator Based on the Artificial Dissipation of Central Convection Schemes	12
2.2	Application of the Error Indicator	14
3	Grid Improvement Strategies and Techniques	18
3.1	Enlargement of the Hexahedra Layer via Grid Manipulation	18
3.2	Local Refinement of Hexahedra to Improve the Wake Discretisation	24
	References	27

Quantification and Reduction of Numerical Uncertainties by Improvement of the TAU Grid Adaptation Tool and Adjoint Methods . . . 29

Matthias Orlt, Nicolas R. Gauger

1	Introduction	29
2	Improvements Using the Decomposability of Elements	31

- 3 Modification of Adapted Grids on the Base of a Geometrical Element Quality 36
 - 3.1 A Geometric Quality Measure for the Element Types of TAU 36
 - 3.2 Modification of Adapted Grids 40
 - 3.3 Results on Modified Grids 42
- 4 Adjoint-Based Error Estimation and Functional Correction 46
- 5 Error Indication Based on the Adjoint Solution 48
- 6 Conclusions for the Application 51
- References 52

Application of Mesh Modifications and Adjoint Error Estimates 55

S. Albensoeder

- 1 Introduction 55
- 2 Methods 56
 - 2.1 Adjoint Error Estimation Method 56
 - 2.2 Mesh Manipulation 57
- 3 Results 59
 - 3.1 Adaptation by an Adjoint Error Estimate 59
 - 3.2 Mesh Manipulation 65
- 4 Conclusion 69
- 5 Outlook 72
- References 72

Part II Turbulence Modeling

Minimization and Quantification of Errors and Uncertainties in RANS Modeling 77

Tobias Schmidt, Charles Mockett, Frank Thiele

- 1 Introduction 77
- 2 Classification of Errors and Uncertainties 78
- 3 Development of the Sensors 79
 - 3.1 Development of the Error Sensors 79
 - 3.2 Development of the Uncertainty Sensors 84
- 4 Application of the Sensors 92
 - 4.1 Error Sensors 92
 - 4.2 Uncertainty Sensors 94
- 5 Applicability on Mixed Errors 96
- 6 Industrial Application 97
- 7 Field of Application and Its Limits 99
- References 100

Sensor Controlled Zonal RANS-LES Method 101

Benedikt Roidl

- 1 Introduction 101
- 2 Numerical Methods 103
 - 2.1 Large-Eddy Simulation 103
 - 2.2 Sensor Development 103
 - 2.3 Synthetic Turbulence Generation Methods (STGM) 104
- 3 Results 107
 - 3.1 Sensor Validation 107
 - 3.2 Validation of STGM 108
 - 3.3 Oblique Shock on Flat Plate (SWBLI) 114
 - 3.4 DRA2303 Transonic Profile 117
- 4 Conclusion 124
- References 125

Part III Numerical Methods

The Application of Iterated Defect Corrections Based on WENO Reconstruction 129

Alexander Filimon, Claus-Dieter Munz

- 1 Introduction 129
- 2 The Method of Iterated Defect Correction 130
 - 2.1 IDeC for Inhomogeneous Problems 134
- 3 WENO Reconstruction on Unstructured Grids 135
- 4 Numerical Results 138
 - 4.1 Convergence Studies 138
 - 4.2 Application Test Cases 147
 - 4.3 Conclusion 152
- References 153

Part IV Geometry and Deformation

Uncertainties of Numerical Structural Models in the Frame of Aeroelasticity 157

P. Reich, A. Reim, M. Haupt, P. Horst

- 1 Finite Element Models and Analysis Methods Used for Uncertainty Quantification 158
 - 1.1 Parametric Finite Element Model 158
 - 1.2 Quantification of Uncertainty 160
- 2 Part I: Model Uncertainties 163
 - 2.1 Introduction 163
 - 2.2 Design of the Wing Structure 164
 - 2.3 Uncertainties due to Modeling Simplifications 165
- 3 Part II: Stochastic Simulations 171
 - 3.1 Introduction 171

- 3.2 First Order Reliability Method 172
- 3.3 Combination of the FORM-Routine with Fluid-
Structure Interaction Code Library 173
- 3.4 Sensitivity Analysis by a Global Variation in Structural
Parameters 174
- 3.5 Results of the FORM Analysis 177
- 3.6 Conclusions 179
- References 180

A Comparison of Fluid/Structure Coupling Methods for Reduced Structural Models 181

Georg Wellmer, Lars Reimer, Horst Flister, Marek Behr, Josef Ballmann

- 1 Introduction 181
- 2 Coupling Methodology 183
 - 2.1 Flow Solver 185
 - 2.2 Structural Solver 186
 - 2.3 Flow Grid Deformation 186
- 3 Spatial Coupling 187
 - 3.1 Finite Interpolation Element Method 188
 - 3.2 Additional Interpolation Schemes for FIE 190
 - 3.3 Global Spline-Based Interpolation (GSB) 192
 - 3.4 Moving-Least-Squares Interpolation (MLS) 194
 - 3.5 Insertion of Additional Support Points 195
- 4 Error Sources in Spatial Coupling 197
 - 4.1 CFD Mesh Spacing and Load Distribution 197
 - 4.2 Influence of Projection Parameters 204
- 5 Coupled Simulations 212
- 6 Conclusion 215
- References 217

Improved Mesh Deformation 219

Holger Barnewitz, Bernd Stieckan

- 1 Introduction 219
- 2 CFD Mesh Deformation Module 221
 - 2.1 Radial Basis Functions in Mesh Deformation 222
 - 2.2 Algorithm 224
- 3 Wall Distance Module 227
- 4 Base Point Reduction Methods 228
 - 4.1 Equidistant Reduction Method 229
 - 4.2 Weighted Distances 229
 - 4.3 Error Correction 232
- 5 Interpolation Quality Comparison 234
- 6 Applications 237
 - 6.1 Wing Shape Design 237
 - 6.2 Multi-disciplinary Wing Optimization–SFB-401-Wing 238
 - 6.3 Application to Complex Configuration 239

7	Summary	240
	References	242

Part V Stochastic Uncertainties

Statistical Analysis of Parameter Variations Using the Taguchi Method .. 247

A. Wolf, D. Henes, S. Bogdanski, Th. Lutz, E. Krämer

1	Introduction	247
2	The Taguchi-Method	248
	2.1 Orthogonal Arrays	249
	2.2 ANOVA-Analysis	249
	2.3 Parameter Interaction	252
	2.4 Error Determination	252
3	Simulation Setup	253
4	Results	255
5	TAU Implementation	262
6	Summary and Conclusion	263
	References	264

Numerical Methods for Uncertainty Quantification and Bayesian

Update in Aerodynamics 265

Alexander Litvinenko, Hermann G. Matthies

1	Introduction	265
2	Statistical Modelling of Uncertainties	266
	2.1 Modelling of Uncertainties in Parameters	267
	2.2 Modelling of Uncertainties in the Airfoil Geometry	268
3	Discretisation Techniques	269
4	Low-Rank Response Surface	270
	4.1 Update of the Low-Rank Response Surface via Computing the Residual	273
5	Data Compression	274
	5.1 Concatenation of Two Low-Rank Matrices	276
6	Bayesian Update of the Uncertain Airfoil Geometry	276
7	Numerics	277
	References	281

**Efficient Quantification of Aerodynamic Uncertainties Using
Gradient-Employing Surrogate Methods 283**

Dishi Liu

1	Introduction	283
2	Test Case	285
3	Methods	287
	3.1 Quasi-Monte Carlo Quadrature	287
	3.2 Gradient-Enhanced Radial Basis Functions	288
	3.3 Gradient-Enhanced Kriging Method	289

3.4	Gradient-Enhanced Point-Collocation Polynomial Chaos Method	290
4	Results and Discussion	291
5	Summary	295
	References	295
Optimal Aerodynamic Design under Uncertainty		297
<i>Volker Schulz, Claudia Schillings</i>		
1	Introduction	298
2	Aleatory Uncertainties in Aerodynamic Design	298
2.1	Mathematical Description of the Uncertainties	299
2.2	Karhunen-Loève-Expansion	300
3	Robust Shape Optimization Problem	301
3.1	Min-Max Formulations	303
3.2	Semi-infinite Formulations	303
3.3	Chance Constraint Formulations	304
4	Reduction of the Dimension of the Probability Space Using a Goal-Oriented Karhunen-Loève Basis	306
5	Adaptive Sparse Grid for High-Dimensional Integration	307
5.1	Adaptive Sparse Grid	309
6	One-Shot Aerodynamic Shape Optimization and Its Coupling to Robust Design	314
7	Numerical Results	316
7.1	Numerical Comparison of the Introduced Robust Formulations	316
7.2	Numerical Results Considering Geometrical Uncertainties (Testcase RAE2822)	319
7.3	Numerical Study of the Influence of Geometrical Uncertainties (Testcase SFB-401)	330
8	Conclusions	334
	References	336
Author Index		339

Part I
Mesh Generation and Manipulation

Methods and Strategies for the Detection and Management of Grid Induced Uncertainties in Numerical Aerodynamics

E. Mazlum and R. Radespiel

Abstract. For the numerical simulation of complex aircraft configurations discretisation errors around wings and tailplanes as well as in their wakes are a key source of computational uncertainties. Both simplifications and uncertainties in the geometry description as well as improper numerical grids have an influence on the prediction of lift and drag coefficients, whereas the error margins are not well known. For this reason, the Institute of Fluid Mechanics systematically analysed and quantified grid induced uncertainties in the frame of the projekt MUNA. Based on this, strategies and tools have been developed for error detection and improvement of improperly discretised grid regions. For the detection of discretization errors, a method based on the artificial dissipation of central schemes has been developed. Furthermore, two methods have been developed for local grid improvement which follow two different grid manipulation strategies. These strategies are the local deformation of grids and the local refinement of hexahedra. The developed methods and tools have been verified on different test cases.

1 Identification of Sources of Error Induced by Geometry and Discretisation

The grid developer has many tools at his disposal. These methods range from manual procedures to fully automated ones, which allow the operator to mesh complex geometries like complete aircraft configurations including wings, tail planes, flaps and engines, by setting only a few parameters. However, these fully automated

E. Mazlum · R. Radespiel

Institute of Fluid Mechanics, Technische Universität Braunschweig
Hermann-Blenk-Str. 37, D-38108 Braunschweig, Germany

e-mail: [e.mazlum, r.radespiel}@tu-bs.de](mailto:{e.mazlum, r.radespiel}@tu-bs.de)

meshing tools generally offer only very limited control on the exact topology of the generated grid. To efficiently analyse the uncertainties in the flow solution caused by the topology of the grid, a grid generation tool is needed which allows the user to shape the grid freely.

Especially for analysing the wake discretization, a grid generation tool is needed which gives the operator full control over the grid generation process. A suited grid generator is Gridgen V15 [1] which is why it was used as the preferred tool in this work. In Gridgen grids are generated manually by creating "connectors", "domains" and "blocks". This approach gives many design options for the shape of the grid in the wake region.

Furthermore, Gridgen allows the generation of hybrid grids. Hybrid grids are a combination of fully structured hexahedral layers, and semi or fully unstructured prism or tetrahedral layers, respectively.

1.1 Numerical Method

The 2D airfoil simulations are performed with the Navier-Stokes solver DLR-TAU [2, 3] using hybrid grids. This flow solver is a finite-volume solver for the Reynolds-averaged Navier-Stokes equations on hybrid grids. The convective fluxes can be discretized either with upwind or central schemes, the latter being used in conjunction with scalar artificial dissipation for the present simulations. Time discretization is done implicitly using a backward Euler scheme [4] in connection with a LU-SGS linear solver.

1.2 Analysis in 2D

For the analysis of low speed cases, the airfoil NLF(0)416 by Somers [5] was chosen. This airfoil is a natural laminar flow airfoil with a very thin trailing edge. A wide base of experimental data at different flow conditions is available. These range from $Re = 2 \cdot 10^6$, $4 \cdot 10^6$ and $6 \cdot 10^6$ for $Ma = 0.1$ as well as $Ma = 0.2$, 0.3 and 0.4 for $Re = 6 \cdot 10^6$. This airfoil was chosen because in addition to the original airfoil with the very thin trailing edge, a version with a blunt trailing edge was also measured experimentally. The bluntness was generated by attaching a wedge on the original airfoil. Thus, numerical simulations with different trailing-edge geometries can be validated using the experimental values.

The subsequent flow solutions were obtained for $Re = 4 \cdot 10^6$ at $Ma = 0.1$, using the Spalart-Almaras turbulence model [6]. The transition was fixed according to the locations given in [5]. All versions of the airfoil were discretized using 210 grid points on the upper surface and 150 grid points on the lower surface. Besides changing the topology of the grid at the wake, the number of grid points along the trailing edges was also varied. Parts of this study have been published in [7].

1.3 Uncertainties Induced by Variation of the Trailing-Edge Geometry

In addition to the original NLF(0)416 trailing-edge geometry, an airfoil with a blunt trailing edge was used. Following Somers [5], the blunt trailing edge was obtained by attaching a wedge on the last five percent of the airfoil. With this wedge the trailing edge gets a bluntness of 0.5% of chord length. As a further variant a NLF(0)416 airfoil with a rounded trailing edge was used. This airfoil was derived from the airfoil with the blunt trailing edge by replacing the corners with radii. The airfoils with sharp as well as blunt trailing edges were discretized using both c- and o-type topology. The airfoil with round trailing-edge geometry was only meshed with an o-type grid since no reasonable c-type discretization was possible. In addition, the grid *c-grid*, $4x$ was derived from the initial c-grid by doubling the number of grid points in both coordinate directions. Figure 1 depicts the grid topologies for the respective trailing edge variants. The numerical results obtained on these three grids are shown in Figure 2. Note that the experimental results on the blunt trailing edge are not depicted in this figure, since the experiments show no influence of the trailing edge geometry in this particular case. The blunt trailing edge leads to a higher drag coefficient in general. Due to the modification of the geometry one gets a difference of 2.0 drag counts in c_d . In the experiments the initial airfoil has a trailing-edge separation. Because of this a flow pattern similar to the flow around the blunt trailing edge appears. But in case of the numerical results only attached flow is present for $\alpha = 4^\circ$ to 12° .

Furthermore, one can observe that the blunt trailing edge yields lower lift coefficients across the whole α -range. This can be attributed to the decambering effect of the wedge attached on the airfoil.

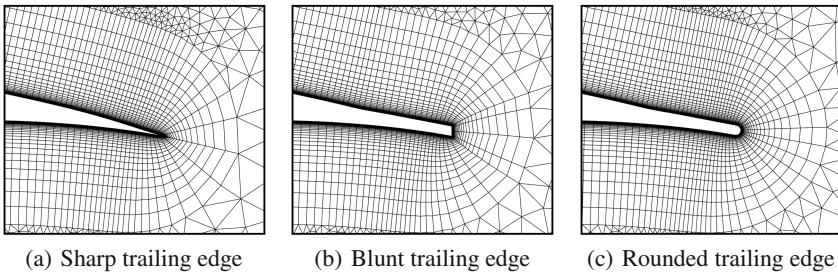


Fig. 1 Trailing-edge geometries

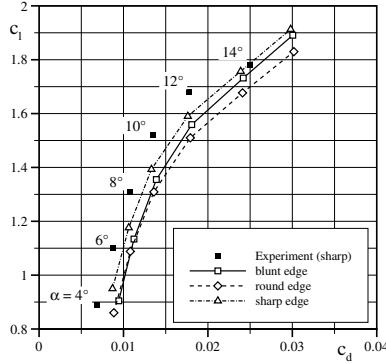


Fig. 2 Polar curves for different trailing edge geometries

1.4 Uncertainties Induced by the Grid Topology

The intention of this study was to analyse the influence of the grid topology on the flow solution. For this purpose a set of four grids was generated. Both trailing-edge variants were discretized in c- and o-shape as shown in Figure 3. Numerical simulations were conducted at $Re = 4 \cdot 10^6$, $Ma = 0.1$. The angle of attack was varied from $\alpha = 4^\circ$ to 14° in steps of two degrees. As closure conditions for the RANS-equations the turbulence model according to Spalart-Almaras has been used. For the c-type grids a grid convergence study was conducted. For this purpose two additional grids were generated by unified coarsening and refinement of the initial grid. The polar curve in Figure 4 shows the result of this study. It compares the corresponding solutions on the c- and o-grids. The polar curves reveal that the grid topology has a significant influence on the aerodynamic coefficients. The variation of the grid topology for the blunt trailing edge yields a variation of about 4.0 to 7.0 drag counts. For the airfoil with the sharp trailing edge the differences vary from 2.50 to 5.50 drag counts. In addition, the differences in c_l and c_d decrease with growing angle of attack for the blunt trailing-edge case. For the airfoil with the sharp trailing-edge no such behaviour is visible.

Analysing the flow solution around the trailing edge is advisable to understand this behaviour. For this purpose Figure 5 shows a detailed plot of the flow solution around the trailing edge at $\alpha = 6^\circ$. Solutions obtained on c- and o- grids are compared. In addition, the wake velocity profile downstream of the trailing edge is shown at three slices. As one can see, there are significant differences between the two solutions. Unlike the o-grid, the c-grid can capture the wake velocity profile and the flow field gradients very well and can preserve them far downstream. The o-grid shows a strong dissipation of the flow features. This can be attributed to the coarse wake discretization on this o-grid. The same statements are valid on c- and o-grids with sharp trailing edges. The variation of topologies mainly shows an

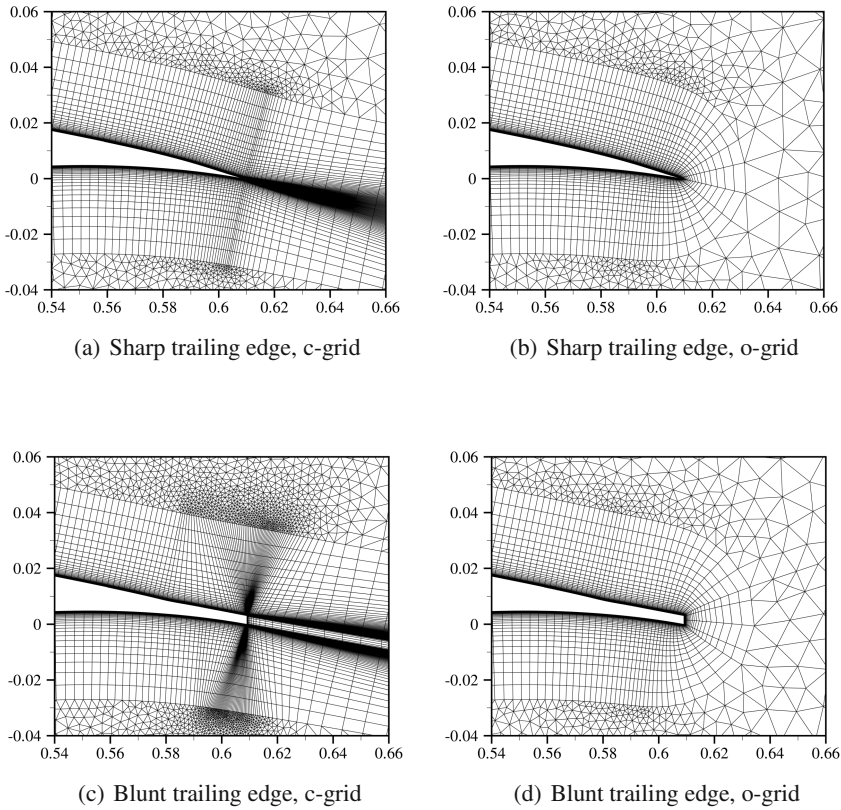


Fig. 3 Grid topology variants with sharp and blunt trailing edges

effect on c_d . The effect on c_l is negligible for attached flows and for flows with small trailing-edge separations.

These results show that meshing trailing edges in o-type topology can be problematic. In general, o-type grids lead to a quick coarsening of the grid downstream of the trailing edge. Especially on airfoils with sharp trailing edges, this coarsening effect is very strong. To reduce this coarsening for o-grids with blunt trailing edges, the number of grid points at the trailing edge has to be increased, whereas for sharp airfoils more grid points have to be shifted towards the trailing edge. However, compared to a c-grid this still does not yield a satisfying solution, since a reasonable resolution of the flow field downstream the trailing edge cannot be achieved even after excessively increasing the number of grid points. The finding, that changing the grid topology has a bigger impact on the flow solution than a change in the trailing-edge geometry is an important conclusion of this analysis.

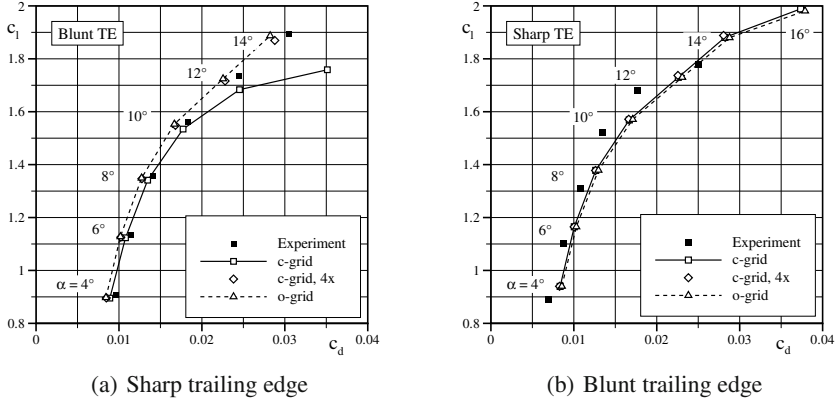


Fig. 4 Influence of different grid topologies and grid resolutions on c_l and c_d of the NLF(0)416 airfoil with different trailing edge geometries at $Ma = 0.1$, $Re = 4 \cdot 10^6$

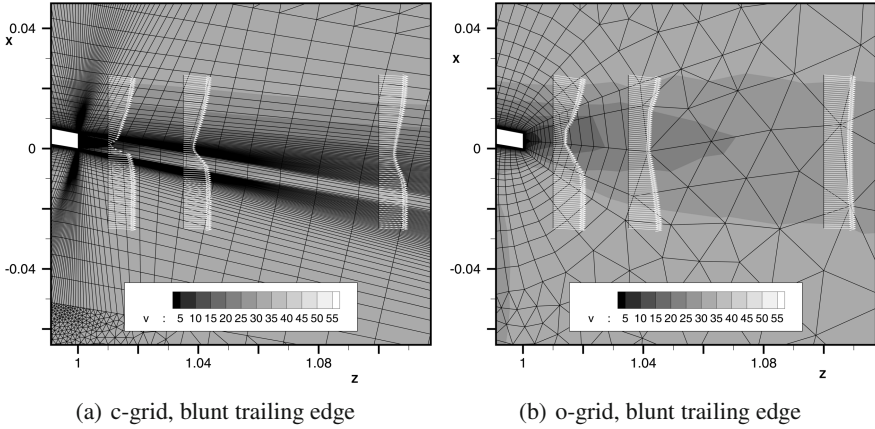


Fig. 5 Flow features of the wake flow field at c- and o-grid, $\alpha = 6^\circ$, $Ma = 0.1$, $Re = 4 \cdot 10^6$

1.5 Effects of the Hybrid Border on the Flow Solution

Hybrid grids are widely used in many scientific and industrial applications. Commonly, these grids have very thin structured layers. But the effects of the size of the structured layer on the flow solution and especially on the aerodynamic coefficients are not well known. Thus, an analysis was conducted for the NLF(0)416 airfoil with sharp trailing edge and c-type mesh. Four grid variants were generated by varying the thickness of the structured layer where the reference thickness was derived from the flat plate boundary layer at $Re = 4 \cdot 10^6$, $Ma = 0.1$. The structured layers were sized with one, two, four and eight times the size of the reference thickness. These four grids, named $1x$, $2x$, $4x$ and $8x$, are shown in Figure 6. To perform a grid

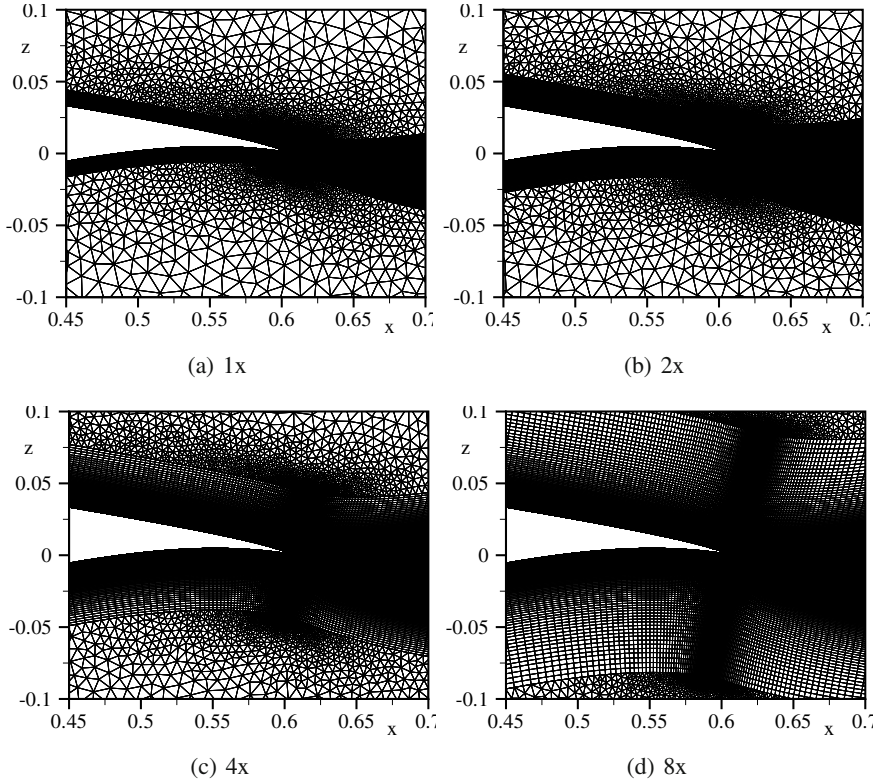


Fig. 6 Variation of the structured layer dimension for the NLF(0)416 airfoil with sharp trailing edge, c-type grid. The dimension is given in multiple of the corresponding flat plate boundary layer thickness.

convergence study, two additional grids were generated by unified coarsening and refinement of the grid δx . Solutions were obtained for $\alpha = 4^\circ$ to 14° in steps of two degrees using the Spalart-Almaras turbulence model. In addition, solutions were obtained at $\alpha = 6^\circ$, 10° and 14° using the SST-Model [8]. The results are shown in Figure 7.

The polar curve shows that δx yields a sufficiently grid converged solution across the whole range of α . The differences to the next finer grid " $\delta x, fine$ " are negligible. The coarse grid " $\delta x, coarse$ " gives slightly different results at $\alpha = 14^\circ$. The polar curves computed with the SA-turbulence-model show that the effect of the structured layer thickness on the lift coefficient is small for attached flow ($\alpha = 4^\circ$ to 12°). The difference in the lift coefficient varies from 1.50 to 3.70 lift counts, while it is increasing with the angle of attack. For $\alpha > 12^\circ$, trailing edge separation occurs, so at $\alpha = 14^\circ$ the differences in the lift coefficients are not negligible anymore. As it will be shown later, this is attributed to the poor resolution of the trailing-edge separation. The differences in the drag coefficients are primarily caused by different

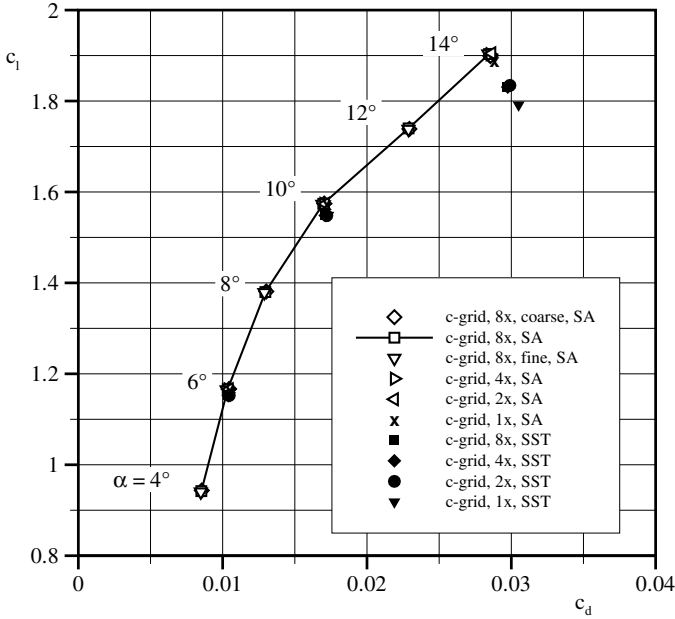


Fig. 7 Influence of the thickness of the structured layer on the lift and the drag coefficients, $Ma = 0.1$, $Re = 4 \cdot 10^6$

levels of resolution of the region surrounding the stagnation point and the suction peak. The thicker the structured layer is, the better it can resolve the flow gradients in that area. The same conclusions are true for the results obtained with the SST-model. For attached flow, differences ranging from 1.50 to 2.60 drag counts are observed. Though, with flow separation at the trailing edge, the SST-model shows a higher sensitivity to the thickness of the structured layer. Here, c_l differs by 3.0 lift counts, whereas c_d differs by 8.20 drag counts. For both turbulence models, the flow solution is moderately affected by the thickness of the structured layer for attached flow. In case of separated flow, a significant influence on both c_l and c_d is observed. These differences can be mainly ascribed to two effects.

The first effect is the influence of the structured layer thickness on the size and shape of the trailing edge separation. Figures 8(a)-8(b) show the flow field solution at the vicinity of the trailing-edge gained with the SA-model. In Figures 8(c)-8(d) the flow field solution obtained with the SST-model are depicted. The eddy viscosity μ_t is shown via a contour plot. In addition, a velocity profile in the separation region is shown. On the grid $1x$ large parts of the boundary layer as well as parts of the free shear layer reach into the unstructured part of the grid. On $8x$ those parts are completely captured by the structure part. A comparison of the plots reveal that the production of μ_t is related to the thickness of the structured layer. Thus, the thickness of the structured layer has an influence on the size of the trailing edge separation. Regarding the case with the SST-model, the influence of the structured

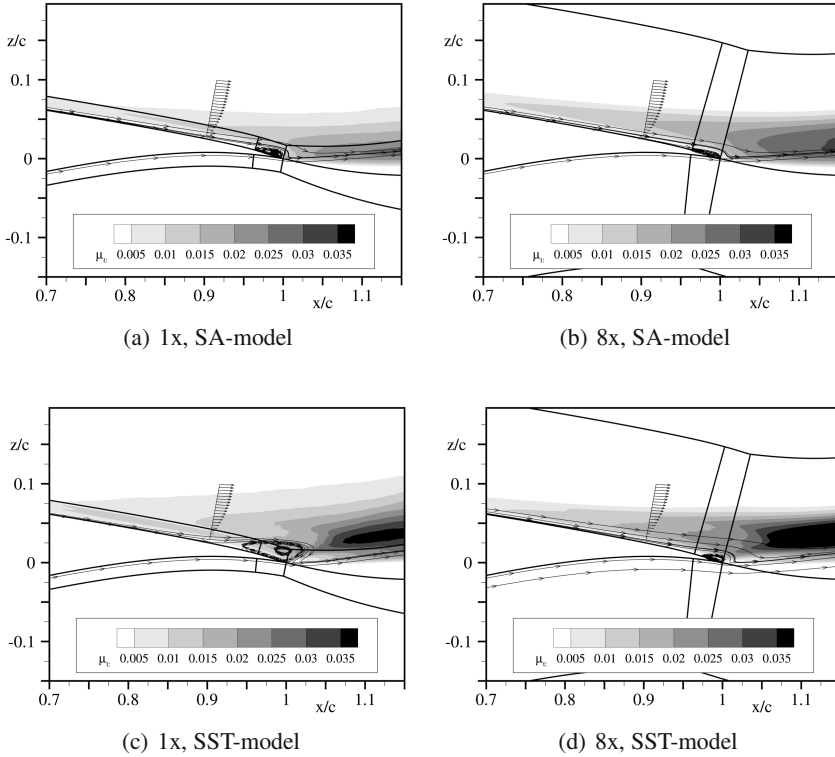


Fig. 8 Flow condition at the trailing edge of the NLF(0)416 airfoil, $\alpha = 14^\circ$, $Ma = 0.1$, $Re = 4 \cdot 10^6$

layer thickness on μ_t is even greater and thus the size of the trailing edge separation is bigger. To further analyse this, the velocity profiles and the rotation $\text{rot}(\mathbf{v})$ at two slices $x/c = 0.3$ and $x/c = 0.9$ normal to the airfoil contour are plotted in Figure 9. At $x/c = 0.3$ the boundary layer is completely within the structured layer, on both grids. Velocities and gradients are continuous and show little difference. At $x/c = 0.9$, the structured part of grid 1x does not cover the boundary layer, the flow separation and the free shear layer completely. Here, the velocity profile has a discontinuity at the hybrid border whereas the velocity gradient shows a kink at $z/c \approx 0.016$. Taking into account that the turbulence equations are solved using the velocity gradients, the influence of the hybrid border becomes obvious.

The second effect is the difference in c_d caused by the erroneous prediction of the trailing edge separation as well as the poor resolution of the flow gradients at the vicinity of the suction peak.

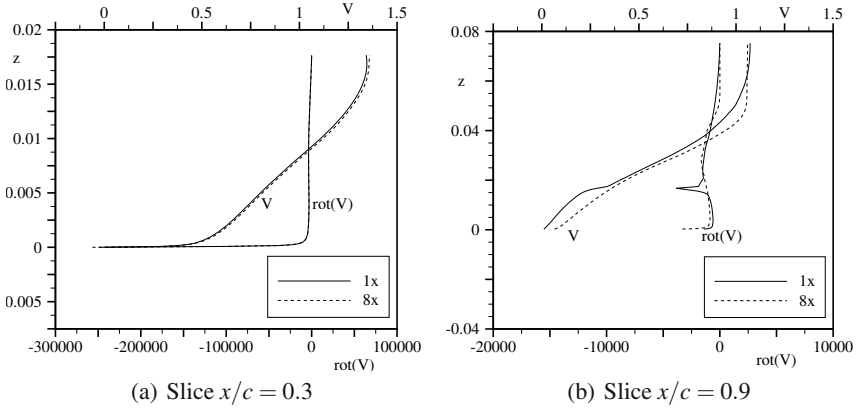


Fig. 9 Plots of the boundary layer velocity curve $V = |\mathbf{v}|$ as well as the boundary layer velocity gradient $\text{rot}(\mathbf{v})$ at two slices normal to the airfoil geometry

2 Methods for the Detection of Improperly Discretised Grid Regions

2.1 Error Indicator Based on the Artificial Dissipation of Central Convection Schemes

The concept of artificial dissipation was originally developed for Euler solvers used to compute inviscid flows. Since inviscid flows have no natural damping mechanism, dispersive error terms can cause oscillations in the solution. To damp those oscillations and reduce the dispersive error an artificial dissipation is introduced. The first artificial dissipation model that incorporates a linear combination of second and fourth order difference dissipation terms was introduced in [9].

For viscous flows, the Navier-Stokes equations provide physical terms that contain natural dissipation effects. However, those dissipative terms are only significant in the viscous shear layer. They are insignificant in flow regions that show characteristics of inviscid flows. Thus, in practice, artificial dissipation terms still have to be introduced for Navier-Stokes computations in order to stabilize the flow in regions with inviscid behaviour.

The artificial dissipation is added to the internal fluxes across the cell faces by modifying the governing equations. Considering the flux $\mathbf{Q}_F^{F,c}$ across a finite volume face F be

$$\mathbf{Q}_F^{F,c} = \frac{1}{2} [\mathbf{F}_r(i) + \mathbf{F}_l(i)], \quad (1)$$

the artificial dissipation is introduced by adding the term $-\frac{1}{2}\tilde{\alpha}\mathbf{D}$ to the RHS of equation (1) which yields

$$\mathbf{Q}_F^{F,c} = \frac{1}{2} [\mathbf{F}_r(i) + \mathbf{F}_l(i)] - \frac{1}{2} \tilde{\alpha} \mathbf{D}.$$

For Navier-Stokes solutions on highly stretched structured meshes, different scalings of the artificial dissipation term both in streamwise and normal direction within the region of viscous flow are needed. For unstructured meshes, directional scaling is difficult to achieve since no mesh coordinate line exists. In order to obtain an adequate scaling of the dissipation for highly stretched portions of the grid, the strategy described in [10] is followed. The scaling factor $\tilde{\alpha}$ is given as

$$\tilde{\alpha} = \lambda_F^c \frac{4\Phi_{F,i}\Phi_{F,j}}{\Phi_{F,i} + \Phi_{F,j}},$$

with

$$\lambda_F^c = |\mathbf{v}_F \cdot \mathbf{F}| + a_f \cdot \mathbf{F}$$

being the maximum eigenvalue of the flux jacobian for the face F . The terms

$$\Phi_{F,i} = \sqrt{r_{F,i}} \quad \text{and} \quad \Phi_{F,j} = \sqrt{r_{F,j}}$$

are necessary to avoid excessive local numerical dissipation in cases of meshes with high-aspect-ratio cells. The term

$$r_{F,i} = \frac{1}{2} \max \left[0; \frac{\lambda_i^c - \lambda_F^c}{\lambda_F^c} \right]$$

relates the size of λ_F^c across the Face F to the total eigenvalue λ_i^c integrated over the entire control volume surrounding P_i , where

$$\lambda_i^c = \sum_{k=2}^n |\mathbf{v} \cdot \mathbf{F}| + a_F \cdot |\mathbf{F}|,$$

and n is the number of surrounding faces. The corresponding terms λ_j^c and $r_{F,j}$ can be defined analogously.

The artificial dissipative flux across the dual face F corresponding to the edge connecting P_i and P_j is given by

$$\mathbf{D}_{i,j} = \epsilon_F^{k(2)} (\mathbf{W}_i - \mathbf{W}_j) - \epsilon_F^{k(4)} (\nabla^2 \mathbf{W}_i - \nabla^2 \mathbf{W}_j), \quad (2)$$

where \mathbf{W} is the vector containing the conservative variables ρ , $\rho \mathbf{v}$ and ρE . The amount of artificial dissipation added to the scheme is controlled by the coefficients

$$\epsilon_F^{k(2)} = k^{(2)} \max(v_i, v_j) \cdot s_{c2}$$

and

$$\epsilon_F^{k(4)} = k^{(2)} \max(0, k^{(4)} - \epsilon_F^{(2)}) \cdot s_{c4},$$

where

$$v_i = \left| \frac{\sum_{j \in N(i)} (p_j - p_i)}{\sum_{j \in N(i)} (p_j + p_i)} \right|$$

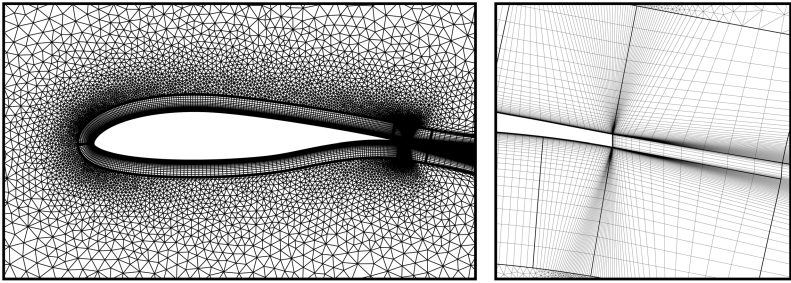
represents a shock switch, where s_{c_2} and s_{c_4} contain some anisotropy corrections and $N(i)$ is the set of neighbours of i . The factors s_{c_2} and s_{c_4} are introduced in order to avoid a dependency of the dissipation on the number of neighbours. The constants $k^{(2)}$ and $k^{(4)}$ are generally user defined parameters in the range of $\frac{1}{4} \leq k^{(2)} \leq \frac{1}{2}$ and $\frac{1}{64} \leq k^{(4)} \leq \frac{1}{32}$. Since the amount of added dissipation is scaled with $\tilde{\alpha} \sim \lambda_F^c$ where for $|\mathbf{F}| \rightarrow 0$, λ_F^c goes to zero, grid converged solution are independent of the dissipation.

In principle this added dissipation has no physical means and thus can be regarded as an unphysical term that introduces an error to the solution. For this reason, the approach presented here regards the added artificial dissipation of central schemes as a measure for the discretization error. Out of the dissipative fluxes added to the scheme, the contribution $d_{\rho E}$ to the energy equation is used as a measure for the discretization error. This contribution has been chosen due to the fact that all key flow variables are represented by the energy equation. The amount of added specific artificial dissipation per cell volume that is being calculated during the evaluation of fluxes is stored and given out as a separate variable of the flow field solution and thus can be evaluated during the post processing. The amount of added artificial dissipation scales with the element face size and it is sensitive to the skewness and the gradients of the adjacent flow field. It is noted here that a skewed cell has high added artificial dissipation as this is based on the cell volume.

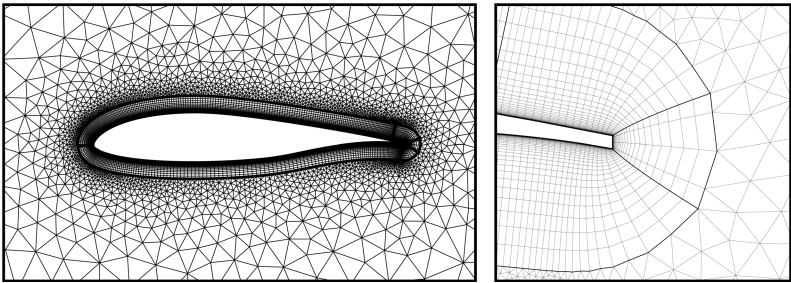
This error indication method does not give an absolute value for the discretisation error since the exact solution of the flow problem is unknown. In fact its intention is to guide the grid developer in deciding where to further improve the grid.

2.2 Application of the Error Indicator

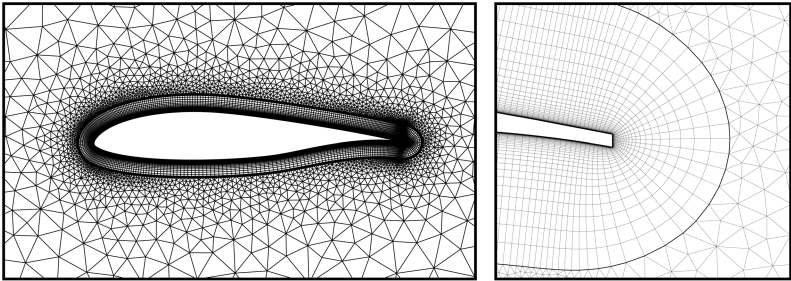
The verification of the error indicator was conducted on six grids. The first four of them are grids of the NLF(0)416 airfoil with a blunt trailing edge. The grid *CBTE* in Figure 10(a) is the reference grid which is of c-type. The grid *HK-Grob* in Figure 10(b) is an o-grid with a coarse trailing-edge discretization. The grid *HK-Fein* shown in Figure 10(c) has a finer trailing-edge discretization. This was achieved by increasing the number of grid points along the trailing edge. The grid *HK-EV* in Figure 10(d) is a further improved version of the previous grid. The number of grid points on the trailing edge was further increased and the sharp trailing edge corners were rounded by the placement of radii. In addition, the structured layer around the first third of the airfoil was enlarged. The last two grids have sharp trailing edges and are of c-type. These are the grids $1x$ and $8x$.



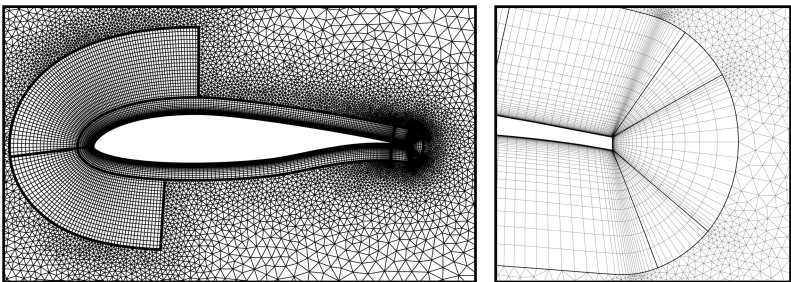
(a) c-grid (*cBTE*)



(b) o-grid (*HK-Grob*) with coarse trailing-edge discretisation



(c) o-grid (*HK-Fein*) with fine trailing-edge discretisation



(d) o-grid (*HK-EV*) with rounded trailing-edge corners

Fig. 10 Grids for the error indicator tests

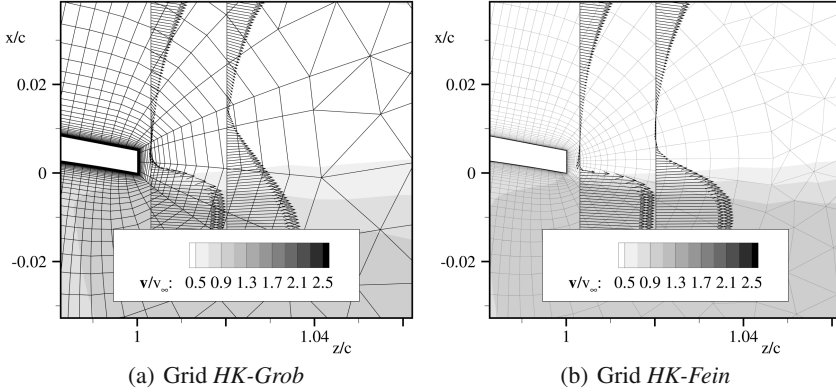


Fig. 11 Flow condition at the vicinity of the trailing edge at two different levels of grid resolution

On these six grids, the solution was gained for $Re = 4 \cdot 10^6$, $Ma = 0.1$ for $\alpha = 4^\circ$ to 14° . For turbulence modelling the SST-model was used. For the grid *HK-Grob* the flow features at the trailing edge are of primary interest. Therefore the features of the flow field at $\alpha = 14^\circ$ are shown in Figure 11(a). At this angle of attack a trailing edge separation is present. Velocity profiles downstream of the trailing edge are shown along two stations with $x = \text{const}$. Due to the o-type grid the resolution in the wake rapidly coarsens which explains why the velocity gradients vanish correspondingly. At $\alpha = 14^\circ$, *HK-Grob* deviates by 4.50 lift counts and 12.0 drag counts from the lift and drag coefficients obtained on *cBTE*.

In case of *HK-Fein*, shown in Figure 11(b), the resolution at the trailing edge is finer which leads to a better resolution of the velocity profiles downstream. To test the applicability of the error indicator on poorly resolved trailing edges, it was applied to *HK-Grob* and *HK-Fein*. The error indicator outputs are compared in Figures 12(a) to 12(d).

Areas that have a small contribution to the discretization error are of light colour whereas areas that have a high contribution are dark coloured. Compared to *HK-Grob* the grid *cBTE* has low error values across the whole wake region. The grid *HK-Grob* is showing a higher level of indicated error along the wake flow field, where the error magnitude increases close to the trailing edge. Due to the free shear layers high gradients at the vicinity of the trailing edge are present. These gradients are poorly resolved by the present coarse grid. Especially at the sharp trailing-edge corners the flow physics yield very high gradients which lead to high values of added artificial dissipation and thus to a high indicated discretization error. As one can see in Figure 12(c) and 12(d) a slight refinement of the wake region by increasing the number of grid points along the trailing edge leads to a significant decrease of the indicated discretization error. However, regions with high error values still persist especially at the lower sharp corner of the trailing edge. At this corner the slow trailing edge flow joins the much faster flow of the lower surface. As a consequence,

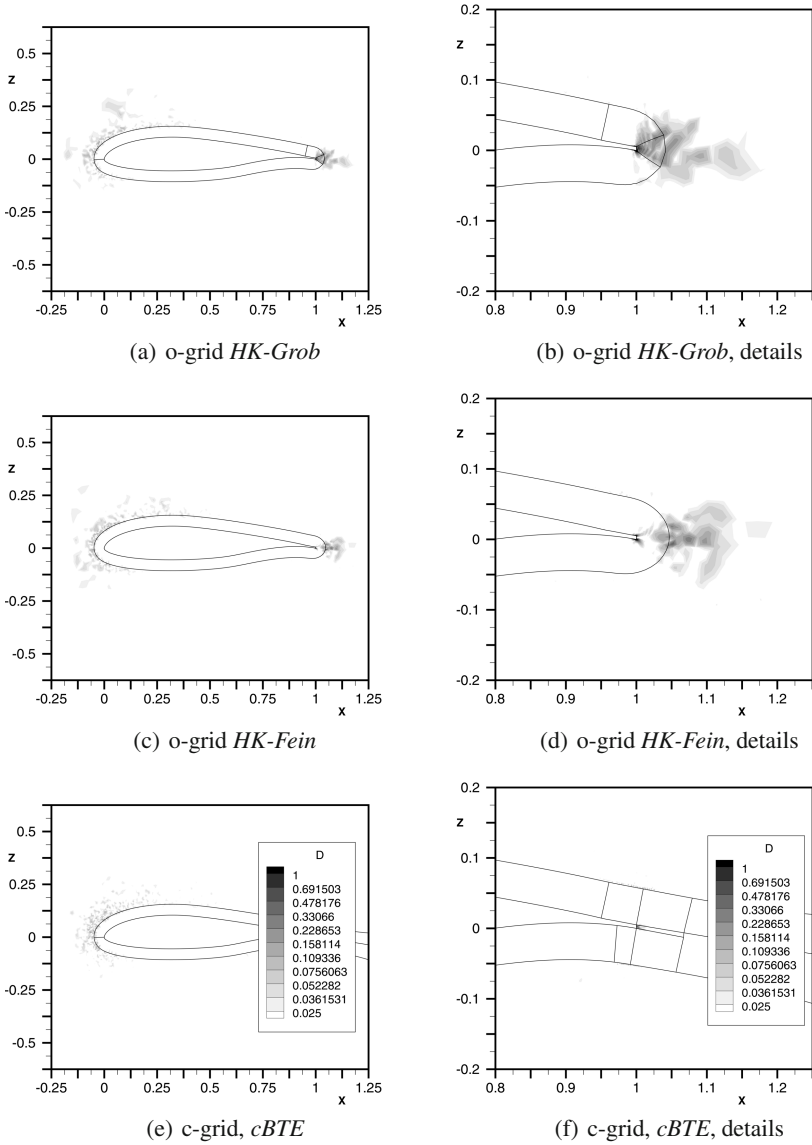


Fig. 12 Error indicator results on test grids

high gradients occur which explain the high indicated error values. Furthermore, the plots reveal that on both *HK-Grob* and *HK-Fein* errors in the unstructured parts of the wake flow and near the leading edge are indicated. This is also attributed to the fact, that the unstructured grid can not sufficiently resolve flow gradients.

A close view of the grid at the trailing edge reveals that the corner cells are highly skewed. This skewness is a trade-off to the sharp trailing edge corners. In

order to improve the mesh around the corners, the grid *HK-EV* in Figure 10(d) was generated.

In Figure 13(a) the output of the error indicator is shown for this improved grid. Figure 13(b) gives a more detailed depiction of the trailing edge region. Comparing the results for *HK-EV* and *HK-Grob* reveals that a significant reduction of the indicated error could be achieved by this grid generation techniques. This is pointed out by the output of the error indicator at the corners in Figure 13(c) and 13(d). In particular the error contribution of the grid at the first third of the airfoil was reduced by enlarging the hexahedral part of the grid as well as through the improvements made near the trailing edge.

Optimizing the trailing edge discretization yields an improvement of 3.0 lift counts and 9.0 drag counts, thus reducing deviations in c_l and c_d to 1.50 lift counts and 3.0 drag counts, respectively (deviations from *HK-Fein* to *cBTE*). Enlarging the structured layer at the first third of the airfoil yields a further reduction of 2.0 drag counts.

In Figure 14(a) and 14(b) the results of the error indicator on the grids $1x$ and $8x$ with sharp trailing edge are shown. Like in the previous cases, regions of increased indicated error are highlighted at the grid region covering the first third of the airfoil. However, the high indicated errors down- and upstream of the trailing edge are of particular interest for this case. These errors are mainly caused by the gradients of the boundary layer and the free shear layer, respectively. Compared to the reference grid $8x$ the grid $1x$ with the thin structured layer has an error of 4.0 lift counts and 8.0 drag counts.

The near-wall velocity profiles as well as the wake velocity profiles are depicted at seven stations along x . At the left-most station (A) the boundary layer is still within the structured grid part. At the following slices (B) to (D) the boundary layer is growing into the unstructured grid part, so the indicated error is growing as well. The indicated error value has a high magnitude at regions with high gradients (C) to (E). In the wake the error values are fading out since the gradients disappear (cf. F and G). On the grid $8x$ in Figure 14(b), no errors can be observed.

3 Grid Improvement Strategies and Techniques

3.1 Enlargement of the Hexahedra Layer via Grid Manipulation

The task of the first grid manipulation tool is to enlarge the structured layer in order to move the hybrid border out of flow regions with gradients. This is achieved by moving the grid points that are located along the hybrid border. These points are moved along wall normal lines of the structured layer. These lines have to be extrapolated into the unstructured grid region in a way that prevents twisting and overlapping of these lines. To assure this, a line extrapolation method similar to those in structured grid generators is used.

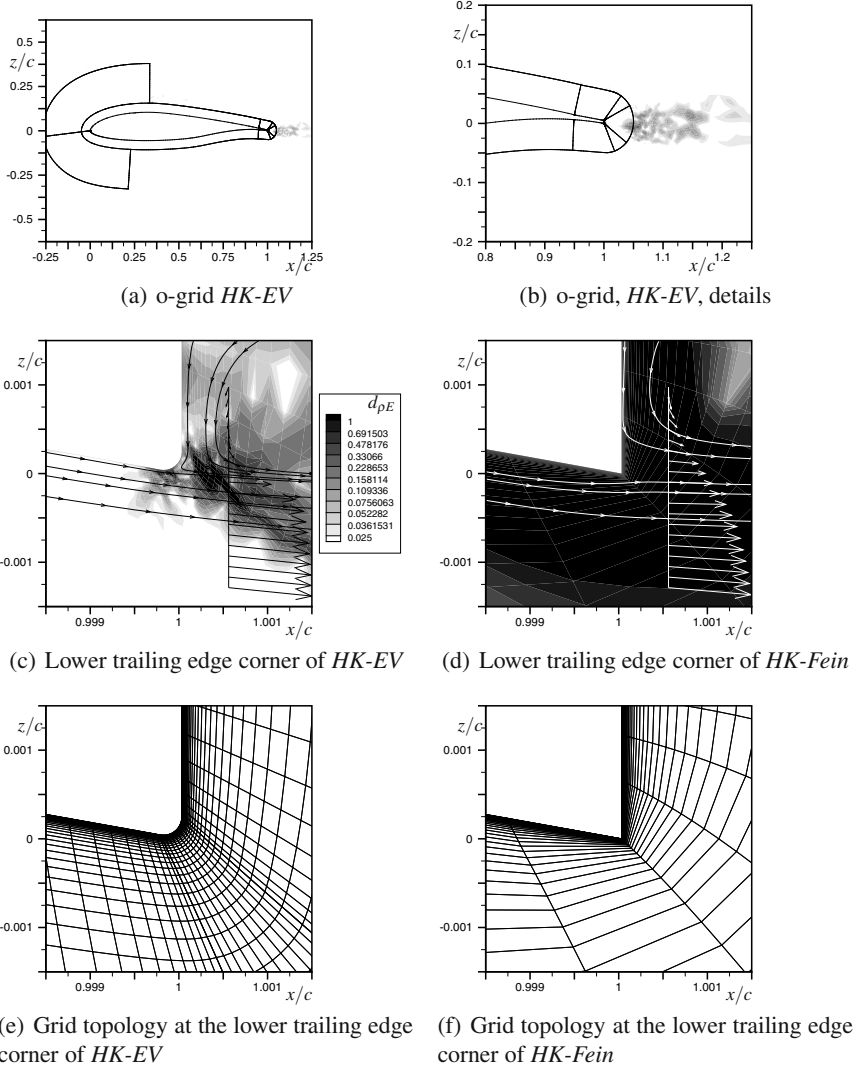


Fig. 13 Error indicator results on *HK-EV*

In the following the proceedings of the tool are described in more detail. After loading the grid the tool determines the surface triangles and quadrilaterals as well as the grid points that shape the hybrid border. Thus a new surface geometry constituted by the surface triangles and quadrilaterals is obtained. Based on this surface a grid of user-defined thickness is generated via wall-normal extrusion. This extrusion is performed according to the methods described in [11] and [12]. To ensure a consistent extrusion of the wall lines, the marching vectors of the extrusion are

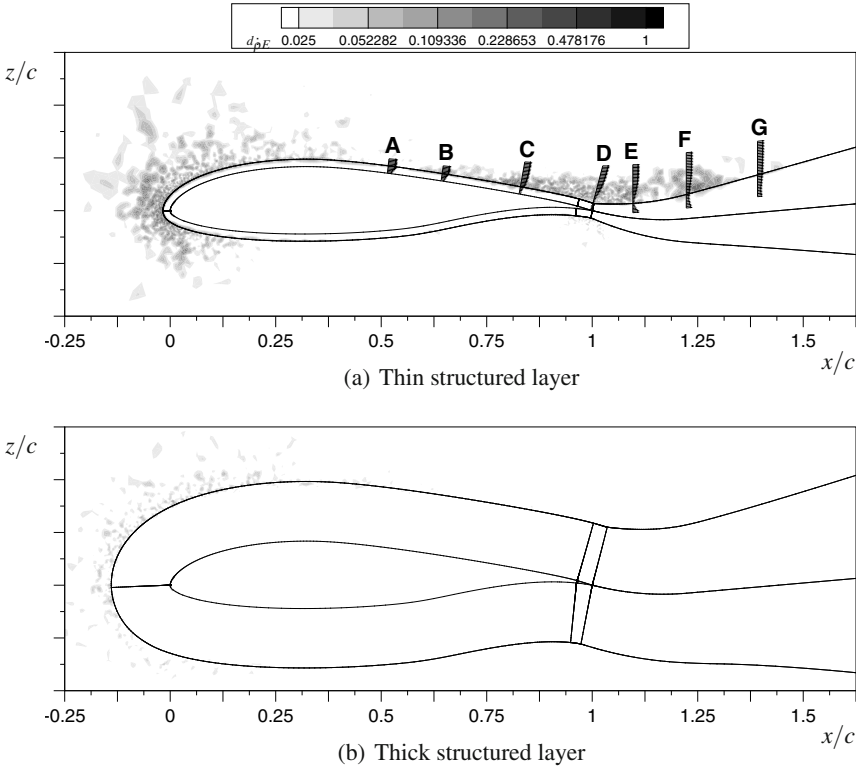


Fig. 14 Error indicator results on different dimensions of structured layers

smoothed [13] in a way that ensures that the visibility cone condition [12] is not violated.

The newly generated points form a new surface geometry on which the extrusion process is repeated. These steps are repeated until a certain number of extrusion layers have been obtained. The resulting grid points are used to define the extrapolated wall lines. Along these wall lines, the new coordinates of the corresponding grid points which shape the hybrid border are moved. To avoid a steep transition from the undeformed to the deformed region, neighbouring points up to a certain degree are included in the movement. This transition is based upon a hyperbolic tangent. After the coordinates of each affected hybrid border grid point have been determined, the points lying below are redistributed. Thereby, discontinuities in the grid point distribution along wall lines are prevented. This redistribution is performed via an area hyperbolic sinus function according to [14]. The coordinates of the affected grid points are stored in a binary file. The manipulation of the grid is performed by a modified version of the DLR-TAU-Code's deformation tool [15] which uses this binary file.

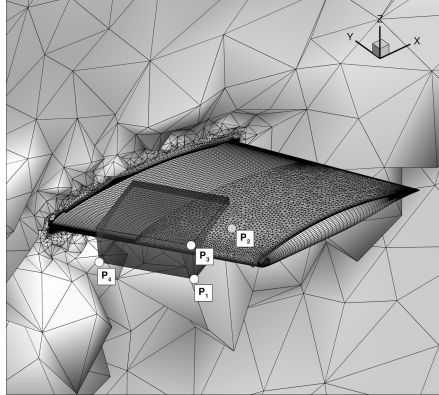
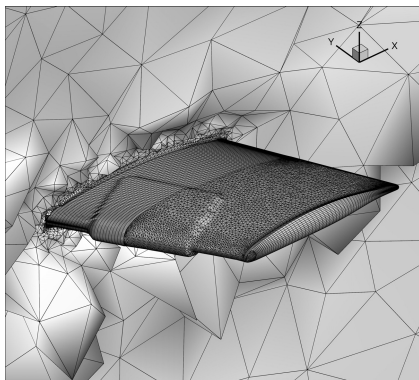


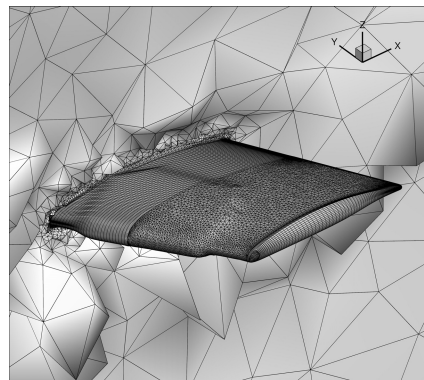
Fig. 15 Illustration of the surface mesh as well as the box for selecting the deformation region

To demonstrate the method, the 3D-grid of the DLR-F6-wingtip with 1.2 million grid points, in Figure 15 is used. Grid lines on the geometry surface are shown by fading out portions of the volume grid. Furthermore, the box which marks the region to be deformed is plotted. This box is defined by the four coordinates $P_1(x,y,z)$ to $P_4(x,y,z)$. Multiple overlapping boxes can be defined.

The result of the deformation is illustrated in Figure 16(a) by showing the prism and hexahedral layers. The enlarging effect can clearly be seen. In this case, neighbourings points of fifth grade have been involved to the deformation. This yields a steep and unsmooth transition from the undeformed to the deformed grid region. To get a smoother transition the whole manipulation process was rerun with neigh-



(a) Neighbours of 5th grade involved in point movement



(b) Neighbours of 15th grade involved in point movement

Fig. 16 Geometry of the deformed hybrid border surface

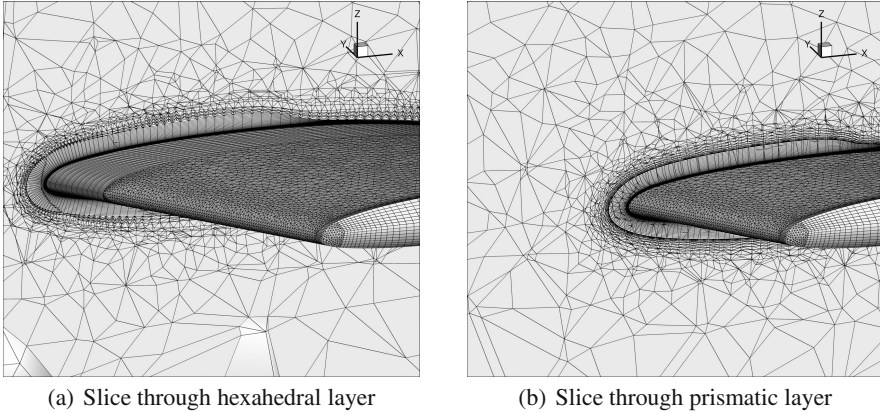


Fig. 17 Slice plane through the deformed grid

bours of 15^{th} being involved. As shown in Figure [16\(b\)](#), this yields a much smoother transition.

For better illustration, the prismatic and hexahedral layers are shown at two slice planes through the volume grid in Figures [17\(a\)](#) and [17\(b\)](#). These Figures show that the transition from the undeformed region to the deformed region is smooth.

In summary, the tool is capable of enlarging the structured layer of a hybrid grid by moving grid points. The degree of enlargement is closely related to the quality of the surface elements forming the hybrid border. If a badly shaped surface element is present at the hybrid border i.e. an element with very high skewness, the ability of the tool to enlarge the structured layer at that location may be limited. However in regions with good element quality, very large deformations are possible.

3.1.1 Application and Results

To improve the grid $1x$ the grid enlargement tool was applied to a portion of the structured layer surrounding the airfoil and the wake. The initial grid was enlarged by the factors two, four and six in each case yielding three additional grids. The resulting grid for an enlargement factor of six is shown in Figure [18\(a\)](#) and compared to the initial grid in Figure [18\(b\)](#). Flow solutions were obtained on these grids. The results are shown in the polar curves plot in Figure [19](#). The polar curve shows that an enlargement by factor two affects the prediction of lift coefficients by about 85% by reducing the difference to the reference grid $8x$ from 4.0 lift counts to 0.60 lift counts. The drag coefficient is affected by a change of about 44% by reducing the difference to $8x$ from 9.20 drag counts to 5.10 drag counts.

Further enlarging the structured layer by factor four and more reverses the improvement as revealed by the polar curve plot. This has to be attributed to the fact that a stronger enlargement reduces the effective resolution in the structured layer,

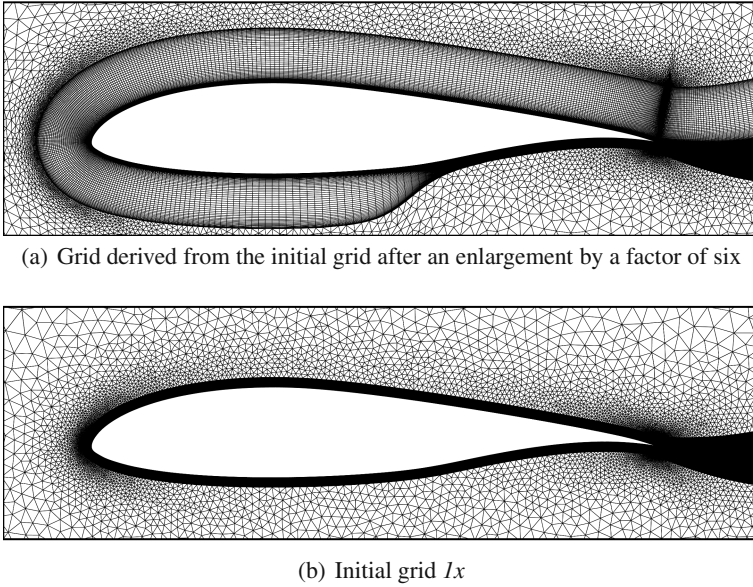


Fig. 18 Initial grid 1x compared to the grid with enlarged structured layer by a factor of six

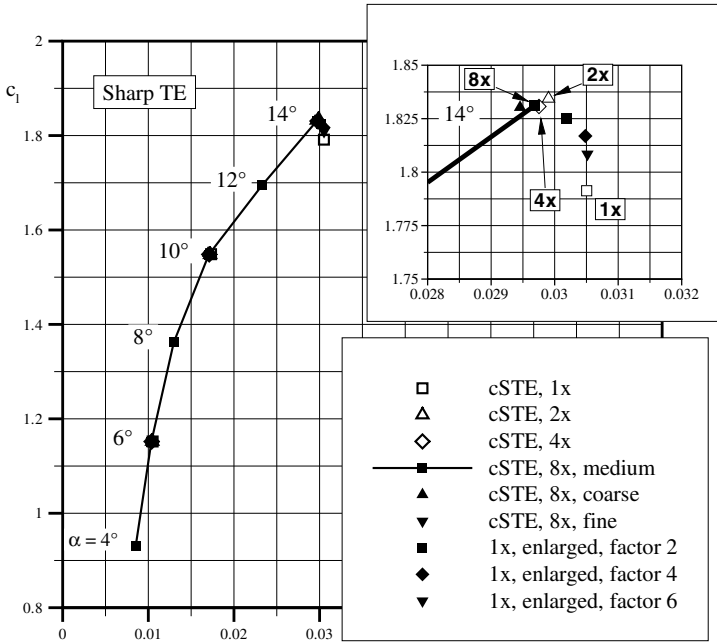


Fig. 19 Effect of the enlargement of structured layers on polar curves

since the enlargement does not introduce additional points but redistributes present points over a wider area.

3.2 Local Refinement of Hexahedra to Improve the Wake Discretisation

3.2.1 Key Features of the Local Hexahedra Refinement Method

To locally refine a hexahedral layer, hanging nodes and faces have to be introduced into the grid. This means that the grid conformity has to be given up. If a solver does not handle hanging entities, the grid has to be made conform by decomposing the hanging relations. This is achieved by decomposing hexahedra facing a hanging entity into prisms or into pyramids and tetrahedra. For the TAU-Solver this task is done by the tool `make_conform`.

As a result, parent hexahedra have to be decomposed into child hexahedra in order to achieve a local refinement. A parent hexahedron can be decomposed in either two, four or eight child hexahedra. For a TAU-grid this means that the child hexahedra are introduced into the grid and the parents are deleted. Thereby, the hanging relations are introduced.

Due to limitations of `make_conform` a parent is allowed to faec on up to a maximum of four child hexahedra. Furthermore, the implementation of this TAU-tool is limited to 3D grids.

As a result of this, the following procedure was established for the present local hexahedra refinement tool, called *hexfine*. Hexahedra destined for refinement by user-input are marked. The child hexahedra of all marked hexahedra are generated. These child hexahedra are constituted by the corner points as well as the face and edge midpoints of the parents. To maintain the smoothness of the grid, the curvatures of the initial grid have to be reconstructed. For this purpose, the face, edge and volume midpoints are interpolated via a bicubic or a tricubic spline. The newly generated children are added to the coordinates and point lists of the grid. The parents are removed. So far the procedure yields a grid with hanging entities which is not TAU-conform. In case of 3D grids, no further steps are necessary and the

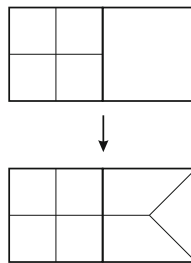


Fig. 20 Schematic illustration of the process to decomposing hanging nodes and faces

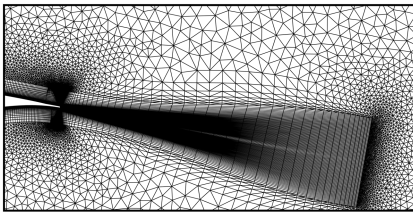
grid can be stored. Applying `make_conform` on the stored grid will give a TAU-conform grid.

In case of 2d grids, the hanging relations have to be decomposed by *hexfine* itself. This is done by decomposing hexahedra facing hanging entities into a prism and two hexahedra (cf. Figure 20). This type of decomposition yields a TAU-conform grid. Running `make_conform` is therefore not necessary for 2D grids.

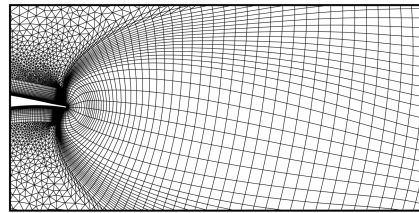
3.2.2 Application of *Hexfine* and Results

To test the method of local hexahedra refinement a set of two main grids were considered. The first one is the grid *cBTE-v2* (including its coarse and fine derivatives) shown in Figure 21(a) which is a c-type grid and has 37,871 grid points. The second grid *oBTE* shown in Figure 21(b) was derived from *cBTE-v2* (including a finer derivative) by remeshing the region around the trailing edge and the wake in o-type. At the trailing edge a structured block was attached to obtain a structured domain reaching far into the wake. It has a total number of 30,127 grid points.

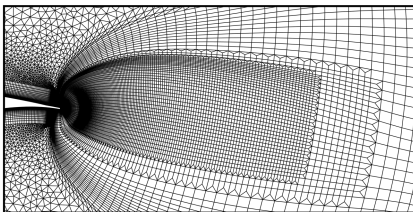
During the first run, the structured wake block of *oBTE* was refined twice. The result of the refinement is shown in Figure 21(c) in comparison to the initial grid in Figure 21(b). This local refinement had the aim to increase the number of grid points in the wake in order to better capture and preserve the flow field gradients. The quadrilaterals at the trailing-edge surface were not affected by the refinement (cf. Figure 22(a) and 22(b)). The number of grid points was raised by about 28% to 38,680 grid points. In a second run, *oBTE* was refined three times. This time, the



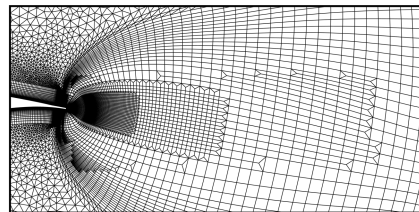
(a) Topology of the c-grid wake region



(b) Topology of the initial o-grid at the wake region, *oBTE*



(c) Topology of the o-grid after refined twice



(d) Topology of the o-grid after refined thrice

Fig. 21 Detailed illustration of the grid topologies at the wake region

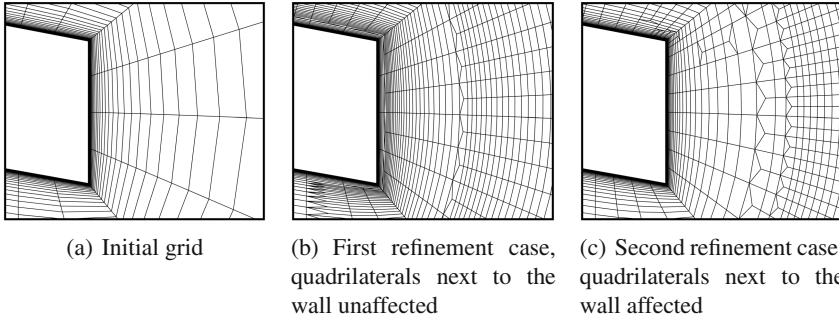


Fig. 22 Comparison of the grid topology at the trailing edge

first refinement affected quadrilaterals up to the wall (cf. Figure 22(c)). The second and third refinement affected a portion of the grid downstream of the trailing edge. The resulting wake grid topology is shown in Figure 21(d). In addition, the grid was refined another three times around the upper corner of the trailing edge with the intention to better discretise the sharp corner. Flow solutions were obtained on

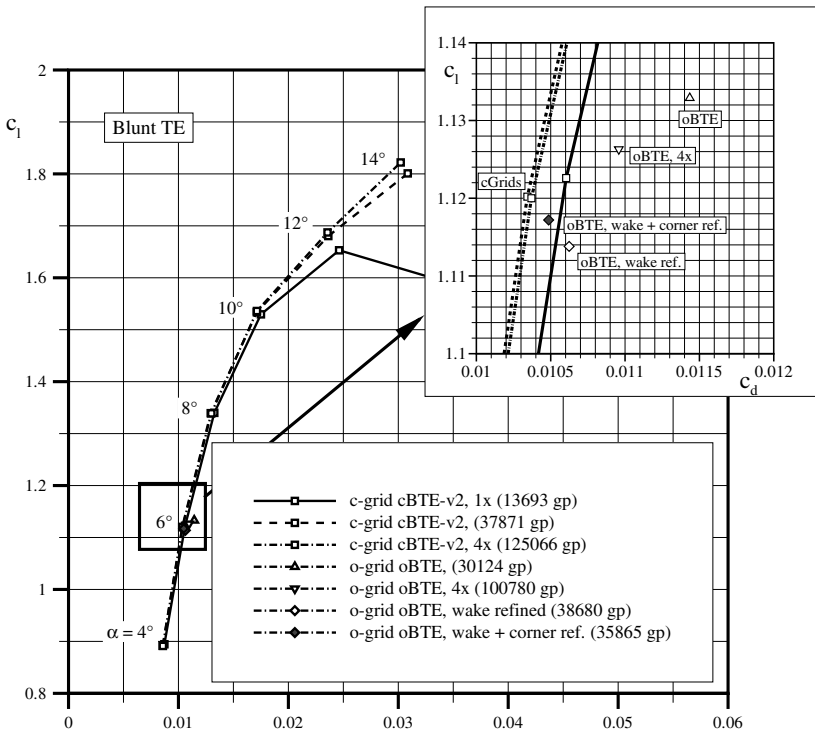


Fig. 23 Results of the local hexahedra refinement on polar curve plots

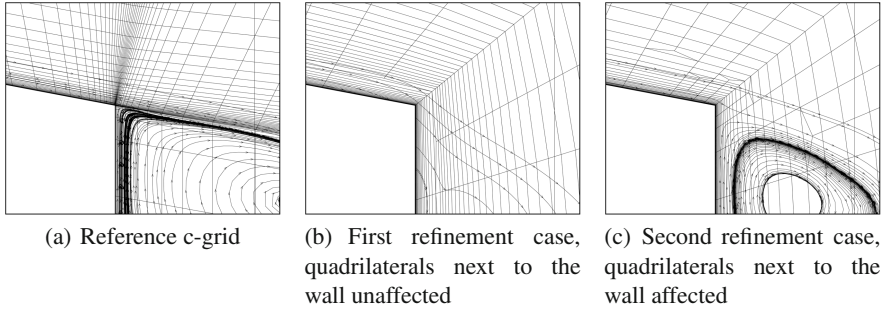


Fig. 24 Grid topology and flow condition at the upper trailing edge corner

these grids. The resulting polar curves are shown in Figure 23. For $\alpha = 6^\circ$, the grid *cBTE-v2*, which is regarded as the reference grid, yields a grid converged solution since the differences in lift and drag to the finer grid are negligible. The coefficients of *oBTE* differ by 1.30 lift counts and 10.90 drag counts to the reference grid.

The *o*-grid with the refined wake reduces the differences to 0.60 lift counts and 2.80 drag counts. This is an improvement of c_d by almost 75%. Analyses showed that refining the wake region a fourth or a fifth time does not yield any appreciable improvements. This reveals that the errors in the flow solution due to a *o*-topology cannot be attributed to a poor wake discretisation alone. In fact, the coarse discretisation of sharp corners with *o*-grids constitutes a further source of error. Analysing the flow features very close to the trailing edge in Figure 24(a) and 24(b) edge reveals, that the way the *o*-grid discretises the trailing edge is not sufficient to capture the sharp corner. This leads to very different flow features at the trailing edge. Figure 24(c) shows that the *o*-grid with refined wake and the additional corner refinement can capture the sharp corner much better, thus leading to less differences in the predicted flow features. As a result, the *o*-grid with the additionally refined upper corner has a difference of only 0.30 lift counts and 1.40 drag counts (cf. polar curve in Figure 23). To sum it up, by well-directed local refinement the error in c_l could be reduced from 1.30 to 0.3 lift counts whereas c_d could be reduced from 10.90 drag counts to 1.40 drag counts. This is achieved by increasing the number of grid points at about only 19%. In contrast, not even a unified grid refinement by doubling the number of grid points in both coordinate directions, which results in a large increase of the total number of grid points by 345.0%, does give a comparable improvement.

References

- [1] Pointwise Inc. Gridgen User Manual, Version 15.13. Forth Worth, Texas (2006)

- [2] Gerhold, T.: Overview of the Hybrid RANS Code TAU, MEGAFLOW - Numerical Flow Simulation for Aircraft Design. In: Kroll, N., Fassbender, J. (eds.) Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol. 89, pp. 81–92. Springer (2005)
- [3] Gerhold, T., Galle, M., Friedrich, O., Evans, J.: Calculation of complex 3D configurations employing the DLR-TAU-Code, AIAA-1997-0167 (1997)
- [4] Dwight, R.P.: Efficiency Improvements of RANS-Based Analysis and Optimization using Implicit and Adjoint Methods on Unstructured Grids. Ph.D. thesis, University of Manchester, Manchester (2006)
- [5] Somers, D.M.: Design and Experimental Results for a Natural-Laminar-Flow Airfoil for General Aviation Applications. NASA Technical Paper 1861, Langley Research Center, Virginia (1981)
- [6] Spalart, P.R., Allmaras, S.R.: A One-Equation Turbulence Model for Aerodynamic Flow, AIAA Paper 92-0439 (1992)
- [7] Probst, A., Mazlum, E., Radespiel, R.: Investigation of computational uncertainties of airfoil flow phenomena close to trailing edges. In: RTO-Symp. on: Computational Uncertainty in Military Vehicle Design, RTO-MP-AVT-147, Athens (2007)
- [8] Menter, F.: Two-equation turbulence models for engineering applications. AIAA Journal 32(8), 1598–1605 (1994)
- [9] Jameson, A., Schmidt, W., Turkel, E.: Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes, AIAA 1981-1259 (1981)
- [10] Mavriplis, D.J., Jameson, A., Martinelli, L.: Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes. In: 27th Aerospace Sciences Meeting, Reno, Nevada, AIAA 1989-0120, January 9-12 (1989)
- [11] Kallinderis, Y., Khawaja, A., McMorris, H.: Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries, AIAA Paper 95-0211 (1995)
- [12] Kallinderis, Y., Ward, S.: Prismatic Grid Generation for Three-Dimensional Complex Geometries. AIAA Journal 31(10), 1850–1856 (1993)
- [13] Field, D.A.: Laplacian smoothing and Delauney triangulations, Mathematics Department, General Motors Research Laboratories, Warren, MI 48090-9057, United States of America (1987)
- [14] Vinokur, M.: On One-Dimensional Stretching Functions for Finite-difference Calculations, NASA Contractor Report 3313, University of Santa Clara, Santa Clara, California (1980)
- [15] Gerhold, T., Neumann, J.: The parallel mesh deformation of the DLR TAU-Code. In: 15. DGLR-Fach-Symposium der STAB, 2006-11-29-2006-12-01, Darmstadt, Germany (2006)

Quantification and Reduction of Numerical Uncertainties by Improvement of the TAU Grid Adaptation Tool and Adjoint Methods

Matthias Orlt and Nicolas R. Gauger

Abstract. Within the framework of the project MUNA, several enhancements of the grid adaptation tool of the DLR TAU Code were prepared, implemented and tested. Therefore, various quality aspects of the single elements of the computational grid were investigated and used to modify the adapted grids.

Conditions for the decomposability of the elements are evaluated and used for a more accurate compliance of the point density with the requirement of the error indicator in the grid refinement. Geometrical element quality terms for the element types of the TAU Code are derived from the known mean-ratio element quality measure for a simplex and used to avoid the worst shaped elements in an adapted grid.

In addition, the TAU adaptation was extended to use the sensors provided by the adjoint solver of TAU as an error indicator for a goal orientated grid adaptation. The results are compared to that of a simple differences-based grid adaptation.

Furthermore, the adjoint method was used for the efficient quantification of uncertainties in the aerodynamic coefficients caused by variations of the parameters of the SAE and the Wilcox- $k-\omega$ turbulence model.

1 Introduction

The goal of any grid adaptation in the solution of partial differential equations is to achieve the highest possible accuracy with the lowest possible effort. Under the assumption that the largest local discretization error affects the solution, the best one can do is to equidistribute the discretization error. So grid adaptation has two main

Matthias Orlt

German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology,
 $C^2A^2S^2E$, Bunsenstr. 10, D-37073 Göttingen, Germany
e-mail: matthias.orlt@dlr.de

Nicolas R. Gauger

Head of Computational Mathematics Group, Department of Mathematics and
Center for Computational Engineering Science (CCES), RWTH Aachen, (formerly DLR)
e-mail: gauger@mathcces.rwth-aachen.de

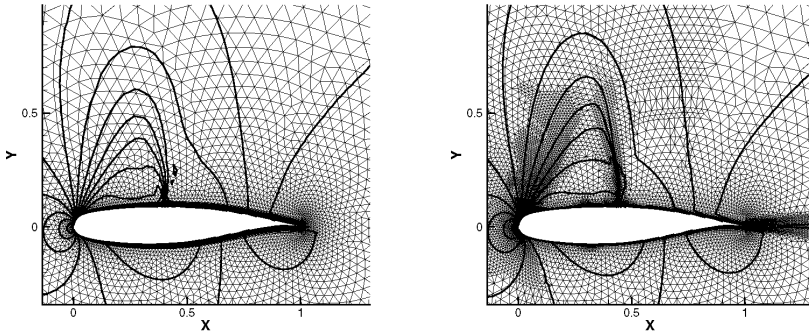


Fig. 1 Example of a grid adaptation around an airfoil, showing the initial grid (left) and the twice adapted grid (right) with different refinement levels and bridging elements between them

steps. The local discretization error is estimated and then the point density is changed. In areas with larger local errors the point density is increased and vice versa.

The TAU adaptation [1] is basically a hierarchical grid re- and derefinement using a special variant of the red-green-method and it works edge orientated. This means edge indicators are determined as numbers specifying the necessity to be refined or derefined for each edge. Depending on the available resources, i.e. a target point number, points are added to or removed from edges. In order to build a new grid from the new points some elements have to be subdivided and some have to be recomposed according to subdivided or rejoined edges.

The advantage of this method is, that it is fast compared to a local remeshing and that it can very accurately fulfill the edge indicator requirements. On the other hand a lot of elements are not refined completely because some refinements have to bridge between various refinement levels in order to get a conform adapted grid. Additionally, the refined area can be very scattered over the grid. This leads to even more bridging elements or nonregular refinements, see Fig. 1.

In order to preserve the semistructured character of the grid near walls in hybrid grids, the vertical edges of the prisms or hexahedra in layers are not subdivided. In areas with chopping, the vertical edges are edges of unstructured elements, too. If such elements are refined it has to be done nonregularly. Moreover there are sometimes badly shaped elements which cannot be refined at all because even a regular refinement would lead to elements which cannot be split into positive tetrahedra parts for the dual computational grid of the TAU solver [1]. Especially large grids for complex geometries tend to include some of such elements, mostly with non-planar quadrilaterals.

The problem of refinement restrictions in conjunction with the element shape around unrefinable edges is addressed in the next Section 2 under the term *element decomposibility*. In this topic, the investigation of the geometric conditions for the

validity of resulting elements enabled for an improvement of the edge refinement algorithm and led to a better compliance of point distribution with the requirements of the error indicator in the adapted grids.

In most cases, the element quality in terms of edge length relation and inner angles of the nonregular bridging refinements is lower than for the corresponding regular refinement. The following Section 3 deals with the element shape of bridging refinements under the term of *geometrical element quality*. Appropriate element quality measures were derived for the element types used in the TAU Code. After a second modification, the TAU adaptation got the option to avoid the nonregular subdivision of badly shaped elements in order to limit the worsening of the element quality in conjunction with bridging refinements.

The last topic is the use of the results of the adjoint solver as a sensor for the edge indicator of mesh adaptation. Furthermore, the adjoint method is used for the efficient quantification of uncertainties in the aerodynamic coefficients caused by variations of the parameters of the SAE and the Wilcox-k- ω turbulence model. For a description of the work at the adjoint solver and the possibility to consider parts of its results as an error estimator we refer to Section 4. The TAU adaptation related work at this point made the adjoint-based sensor available for the internal edge indicator and enabled for an *adjoint-based adaptation*. Some first tests were done within the project. They are presented and discussed in Section 5.

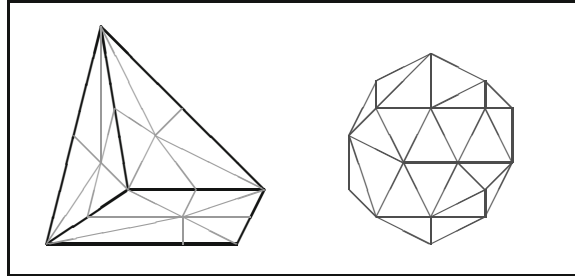
The three modifications include a better error indication, a more accurate adaptation to the required point density and a check of the geometrical element quality. Each of them aims at the reduction of the numerical uncertainty.

2 Improvements Using the Decomposability of Elements

According to the red-green-type of refinement and the edge-oriented methods of the TAU adaptation, a lot of various types of element decompositions are needed for each element type. The *decomposability* of a special element means its geometrical characteristics which determine which subdivisions can be applied to this element. A special subdivision is applicable to an element if the resulting child elements are admissible for the TAU solver. Elements are admissible if they can be split into tetrahedral parts using the midpoints of the element, of the element faces and edges, see Fig. 2. Each of these tetrahedral parts has to have a positive volume, i. e. has to be correctly orientated, because the dual control volumes for the TAU solver are built out of them.

The problem is trivial for tetrahedra. Each tetrahedron with a positive volume has a mid point decomposition into positive tetrahedral parts. Furthermore, each refinement of a tetrahedron resulting from subdivision of some of its edges provides child tetrahedra with positive volume only. So even a very stretched tetrahedron has the full decomposability quality in this sense. This is not true for the other element types. Mainly because of nonplanar quadrilateral element faces, the similarity of the descendants and the initials does not hold as for tetrahedra.

Fig. 2 Split of a pyramidal element into tetrahedral parts, including the element midpoint and the face and edge midpoints (left), and recombination of the tetrahedral parts around a node to a dual control volume of the TAU solver (right)



Pyramids are not only part of bridging refinements between various hexahedra refinement levels, they also bridge between hexahedra and tetrahedra of hybrid grids and at prism sides in case of a varying number of prism layers. Thus pyramids have the most complicated system of subdivisions in the TAU adaptation, see Fig. 3. That is why the following consideration is done for the example of pyramids.

The admissibility conditions for the descendants of a special element subdivision can be expressed in terms of the edge vectors of the initial element. The vector analysis yields the corresponding conditions for the initial element needed to fulfill the decomposability condition for each single subdivision. The results of this investigation are presented in Tab. 1. The notation used in the formulation of the decomposability is introduced in Fig. 4.

In order to see how the exact knowledge about the possible refinements is used, some details of the adaptation algorithm are needed. There are two critical points: the first one is that most grids have some unrefinable edges, at least for the TAU adaptation. That can be the vertical edges of hexahedra and prisms of the structured and semistructured areas for resolving boundary layers. There are badly shaped


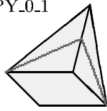
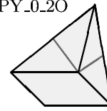

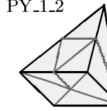

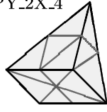
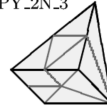






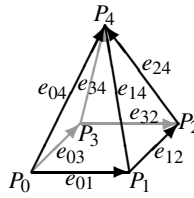
PY_0.0  1 Pyra 0 Tetra	PY_0.1  1 Pyra 2 Tetra	PY_0.20  1 Pyra 4 Tetra	PY_0.4  1 Pyra 6 Tetra	PY_1.2  0 Pyra 9 Tetra
PY_1.4  0 Pyra 13 Tetra	PY_2X.4  0 Pyra 14 Tetra	PY_2N.3  1 Pyra 10 Tetra	PY_2N.4  1 Pyra 12 Tetra	PY_3.4  2 Pyra 11 Tetra
PY_20.0  2 Pyra 0 Tetra	PY_20.2  2 Pyra 4 Tetra	PY_20.4  2 Pyra 7 Tetra	PY_4.4  4 Pyra 8 Tetra	

Fig. 3 System of pyramid subdivisions that is needed for TAU adaptation including the trivial (non-)refinement (PY_0_0) and the regular refinement (PY_4_4)

Fig. 4 Notation of edges and volume parts in pyramids



Notations:

- $V_0 = (e_{01} \times e_{03}) \cdot e_{04}$ Triple products are “local volumes” of corners in some sense,
- $V_1 = (e_{01} \times e_{12}) \cdot e_{14}$
- $V_2 = (e_{32} \times e_{12}) \cdot e_{24}$
- $V_3 = (e_{32} \times e_{03}) \cdot e_{34}$
- $V = \frac{1}{12} (V_0 + V_1 + V_2 + V_3)$ total volume.

Table 1 Decomposability conditions for pyramids

Level	Decomposability conditions	Possible refinements
Pyr-F	$V_i + V_{i-1} \leq 0$ for one $i = 0, 1, 2, 3$	pyramid is not admissible
Pyr-0	$V_i + V_{i-1} > 0$ for all $i = 0, 1, 2, 3$	PYRA_0_0, PYRA_2X_4, PYRA_1_2, PYRA_1_4
Pyr-1	$3V_i + V_{i-1} > 0$ und $V_i + 3V_{i-1} > 0$ for all $i = 0, 1, 2, 3$	PYRA_2N_3, PYRA_2N_4, PYRA_3_4, PYRA_4_4
Pyr-2	$2V_{i+1} - V_i + V_{i-1} > 0$ and $V_{i+1} - V_i + 2V_{i-1} > 0$ and $V_i > 0$ for all $i = 0, 1, 2, 3$	PYRA_0_1, PYRA_0_2O, PYRA_0_4, PYRA_2O_0
Pyr-3	$V_{i+1} - V_i + V_{i-1} > 0$ and $V_i > 0$ for all $i = 0, 1, 2, 3$	PYRA_2O_2, PYRA_2O_4

elements which cannot be refined at all in some grids for complex geometries. Furthermore, depending on the TAU adaptation version, there are elements which have no implemented refinement, e. g. hexahedra apart from the boundary layers or hexahedra which cannot uniquely be assigned to one stack up to now.

The second point is that the initial edge refinement by the edge indicator runs through the grid to some extent. Starting from the initial refinement, additional edges have to be refined due to the red-green method and in order to preserve the layer structure at boundaries. Because not all element types have a subdivision case for each combination of marked edges, additional edges have to be marked in order to find a valid refinement case.

If this ongoing edge subdivision runs onto an unrefinable edge, the adaptation will fail with an invalid subdivision state for an element and crash. Earlier versions of the TAU adaptation used the following method to avoid this situation:

Earlier method:

1. Find unrefinable edges.
2. Mark all edges of a 5 elements deep environment around unrefinable edges as forbidden for initial subdivision.
3. Do the initial subdivision due to the edge refinement indicator for allowed edges only.
4. Loop over all elements:

Subdivide additional edges (including the forbidden and refinable edges) until a valid refinement state is reached.

The depth of 5 not initially refined elements was needed for stability in application examples with repeated adaptation. In this way large parts of the refinable grid area were excluded from adaptation. The preliminary investigation of the decomposability quality enabled the following improvement of the method:

Improved method:

1. Find unrefinable edges.
2. Check the decomposition quality for each element.
3. Loop over all elements:
Mark additional edges as unrefinable until each initial refinement leads to at least one valid refinement state.
4. Do the initial subdivision due to the edge refinement indicator for all refinable edges.
5. Loop over all elements:
Subdivide additional edges until a valid refinement state is reached.

The influence of the improved method was tested with a simulation of the flow around the HIRENASD wing. The initial grid was a SOLAR-grid [8] with 3.12 million points. After the solution was fairly well converged, the grid was adapted with 50% new points using the adjoint-based error indication and the integral drag coefficient as the target functional, see contribution *Adjoint-Based Error Estimation and Functional Correction* of this volume and Section 5. In case of adaptation to a target functional, the accuracy of the result for this functional is obviously a criterion for the performance of the adaptation.

The influence of the various refinement algorithms can be observed in a cut for $y = 0$ near the symmetry plane behind the trailing edge, see Fig 5. The adapted grid resulting from the earlier algorithm has much less points in this area. It can be supposed that this is due to the five elements deep environment of edges forbidden for the initial refinement.

A zoom into the grid around the trailing edge actually shows the unrefinable edges, see Fig. 6. The grid for the boundary layer consists of hexahedra stacks on the wall. They are depicted as quadrilaterals in the cut planes. But there are two stacks on the sides of prisms. These prisms are depicted as triangles at the lower corner of the thick trailing edge in the cut plane in the right side of Figure 5. The prisms alone would not prevent refining the attached hexahedra stacks. But the prisms form two chains along the trailing edge perpendicular to the cut plane and at the other end of

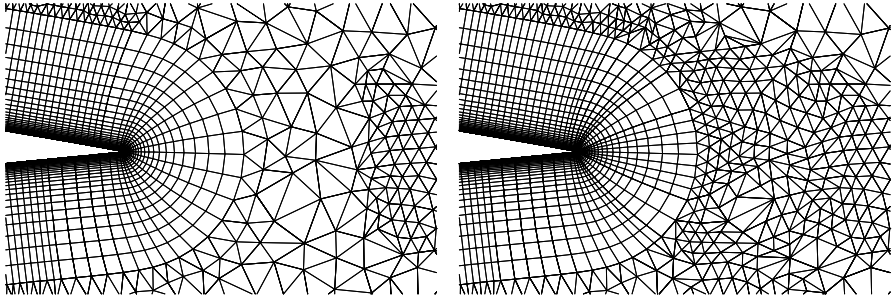
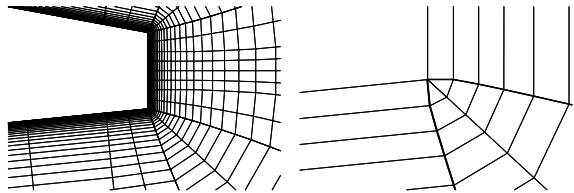


Fig. 5 Comparison of adapted grids, using the old algorithm and a forbidden environment around unrefinable edges (left), and the improved algorithm, performing a preliminary evaluation of the decompsability of the elements

Fig. 6 Zoom into the improved adapted grid of the test example, a weaker zoom (left) and a detailed view showing the prism chains (triangles in the cut) along the lower corner of the trailing edge (right)



the prism chain, at the wing tip, a badly shaped pyramid is attached. Some edges of this pyramid are vertical edges at the same time.

The old algorithm starts at the unrefinable vertical edges, finds that the bad shaped pyramid only can be refined at once. Thus the entire pyramid is unrefinable. Because the TAU adaptation considers the attached prism chain as a stack, the corresponding edge stack would also be refined at once. Hence the algorithm which forbids the environment has to go through the prism chain in one step. Also, the algorithm has to go through the hexahedra stacks at the prism sides in one step. Finally it propagates into the tetrahedra area, affecting some of the neighbouring hexahedra stacks. Unfortunately this affects the whole trailing edge. The new algorithm does not need the five elements deep environment of forbidden edges. Therefore only the hexahedra stacks at the prisms are forbidden for initial refinement and not additionally the neighbour stacks and tetrahedra around them.

The results for the different adapted grids and the initial grid and for a globally refined grid as reference result are given in Table 2. The solution on the globally refined grid is a useful reference value for a solution on a once only locally adapted grid because the globally refined grid has in each region the best state which any local refinement could have regarding the resolution and element shapes.

The comparison shows that the drag is evaluated more accurately on the grid adapted using the improved algorithm. The result of the lift also is approximated better in comparison to the result on the initial grid. Because the target functional of the adjoint-based adaptation was the drag, the better adaptation algorithm does not have to result in a better approximation of the lift coefficient.

Table 2 Results of the test example

value	initial grid	adapted (old)	adapted (new)	global refined
C_L	2.783e-1	2.788e-1	2.787e-1	2.789e-1
C_D	1.181e-2	1.180e-2	1.179e-2	1.171e-2

3 Modification of Adapted Grids on the Base of a Geometrical Element Quality

The background of another new option of the TAU adaptation is that the error estimation, of advanced adjoint-based methods as well as of simple differences-based methods, determines a new point density for the initial grid. On the other hand, the local discretization error depends on the local resolution and probably on the element shape and alignment. So the ideal grid adaptation would adapt the point density without changing or at least without worsening the element shape. This is impossible for a hierarchical conform refinement as the TAU adaptation performs.

The preferable solution was to evaluate the influence of the element shape on the local discretization error and to consider the result when determining the refinement state. However, the attempt to estimate the numerical error by evaluating the numerical fluxes in the control volume or at least the first derivative of a variable for the methods used by TAU lead to very complex formulae and a large diversity of possible element configurations around a point. Because of the low prospect of success this trial was abandoned. In the mean time, a similar problem seems to be solved for the two-dimensional case, using symbolic computations [7].

Partial results of the analytical investigations suggest that the local discretization error is comparatively small for uniform grids. Therefore, it seems to be worthwhile considering a geometrical element quality [2, 4, 12] for replacing the element shape related part of the local discretization error. The main problem of this approach seems to be that the converse argument is not true. Grids or areas with apparently low element quality may provide good dual grids in terms of rectangularity of dual edges and faces.

3.1 A Geometric Quality Measure for the Element Types of TAU

The geometrical quality measure has to be defined for all element types used in TAU, and the choice has to consider the available information and experience about the relation between element shape and discretization error. The high aspect ratio of hexahedra or prisms for example is needed to simulate flows with anisotropic character. The analytic preliminary investigations also confirmed that the aspect ratio of quadrilaterals does not contribute to the element shape related discretization error in an otherwise regular grid.

Without a good reason not to do so the most simple measure is chosen for the unstructured part. A very simple and well known geometrical quality measure for simplices is the *mean-ratio* shape measure, e.g. [4]:

$$Q_{tri} = \frac{2(\lambda_1\lambda_2)^{1/2}}{\lambda_1 + \lambda_2} = \frac{4\sqrt{3}A}{a^2 + b^2 + c^2} \quad (1)$$

$$Q_{tetra} = \frac{3(\lambda_1\lambda_2\lambda_3)^{1/3}}{\lambda_1 + \lambda_2 + \lambda_3} = \frac{12\sqrt[3]{9V^2}}{\sum_{0 \leq i < j \leq 3} e_{ij}^2} \quad (2)$$

This is the ratio of the geometric and the arithmetic mean of the eigenvalues λ_i of the linear transformation to the regular element. The resulting formulae for triangles [1] and tetrahedra [2] have a representation in terms of the element area A and the edge length a, b, c and in terms of the volume V and the edge length e_{ij} , respectively.

Most quadrilateral quality measures suggested in publications are derived from triangle quality measures such that high aspect ratios mean a low quality, e.g. [9]. So they are not applicable to TAU. Here another approach is tried. Instead of decomposing the quadrilateral and applying the triangle formula to the pieces, here the formula is decomposed and the parts interpreted as penalty terms of certain geometrical distortions. The analogue application to quadrilaterals shows how the term has to be modified to get a useful formula for TAU.

By means of the vector analysis the following representation of the mean-ratio quality for triangles can be found:

$$Q_{tri} = \frac{4\sqrt{3}A}{a^2 + b^2 + c^2} = \frac{\sqrt{3}|a||b+c|\sin\phi}{a^2 + \frac{1}{2}a^2 + \frac{1}{2}|b+c|^2} = \sin\phi \cdot \frac{\sqrt{|a|^2 \cdot \left|\frac{b+c}{\sqrt{3}}\right|^2}}{\frac{1}{2}\left(|a|^2 + \left|\frac{b+c}{\sqrt{3}}\right|^2\right)} \quad (3)$$

This can be considered as a decomposition of the mean-ratio quality into a term $\sin\phi$ for the evaluation of distortion and a second term evaluating the stretching of the triangle. The stretching related term again has a mean ratio structure. It is the ratio of geometric and arithmetic mean of $|a|$ and $\left|\frac{b+c}{\sqrt{3}}\right|$ which are equal for a regular triangle. The decomposition into distortion and stretching depends on the choice of the triangle base side. This does not matter because the triangle formula will not be changed and the analogue formula for quadrilaterals which has to be changed will not depend on an arbitrary choice. Figure 7 shows the further proceeding.

The analogue formula considering distortion and stretching in the same way for quadrilaterals [4] does not include deviations from parallelism. So Q_{quad}^* can be considered as quality measure for parallelograms, see Fig. 7 middle part. In order to get a useful quality measure for TAU, the mean-ratio term evaluating the stretching is neglected and a term X for evaluation of the nonparallelism is added.

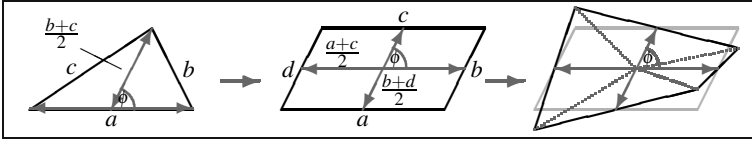


Fig. 7 Steps of derivation of the geometrical quality measure for quadrilaterals: interpretation as stretching and distortion (left), construction of the analogue measure for a parallelogram (mid) and addition of a term to evaluate the deviation from parallelism (right)

$$Q_{quad}^* = \sin \phi \cdot \frac{\sqrt{|a+c|^2 \cdot |b+d|^2}}{\frac{1}{2}(|a+c|^2 + |b+d|^2)} \quad \sin \phi = \frac{|(a+c) \times (b+d)|}{|a+c| \cdot |b+d|} \quad (4)$$

$$Q_{quad} = \sin \phi \cdot X = \frac{2(|a \times (b+d)| \cdot |c \times (b+d)| \cdot |(a+c) \times b| \cdot |(a+c) \times d|)^{\frac{1}{4}}}{|a+c| \cdot |b+d|} \quad (5)$$

The choice of the term X is determined by the mean-ratio principle and the definition of a correct element quality measure. This includes that this measure is zero for a completely collapsed or otherwise distorted element. A measure is needed to evaluate elements of the TAU Code in a reasonable way. So an element has to be considered as completely distorted, if one of its triangle or tetrahedra parts collapses. The definition of X as the mean ratio, i. e. the ratio of geometric and arithmetic mean, of the triangle part areas meets both requirements, see Fig. 7, right part. The resulting quality measure for quadrilaterals Q_{quad} can be expressed by formula (5).

The application of this idea to the three-dimensional case includes a geometrical interpretation and a decomposition of the mean-ratio quality of tetrahedra. In this calculation the terms describing the base triangle can be replaced by the analogue terms for quadrilaterals, see Fig. 8, upper part. In this way a consistent quality measure for pyramids is derived. The quadrilateral quality can be generalized to hexahedra, considering the relation of triangle and tetrahedra quality. An application of this step to triangle bases leads to a quality formula for prisms, see Fig. 8, lower part.

This procedure ensures a kind of consistency between the quality measures of various element types. In this context consistency means, that similar distortions applied to elements of various types lead roughly to the same decrease of the element quality for both element types, see Fig. 9.

This feature is very valuable for the use in adaptation algorithms. It could be used in the investigation of element subdivisions. For example, if a hexahedron of the structured layers is refined to bridging prisms, the prisms roughly inherit the quality of the parent hexahedron, and their quality is decreased by the horizontal stretching which was not relevant for the parent hexahedron. This decrease is nearly independent of the initial quality of the parent.

Fig. 8 Scheme of derivation of the quality measures for the various element types by starting from simplices (grey), adding a term for nonparallelism (lightgrey) and derivation of the remaining (black)

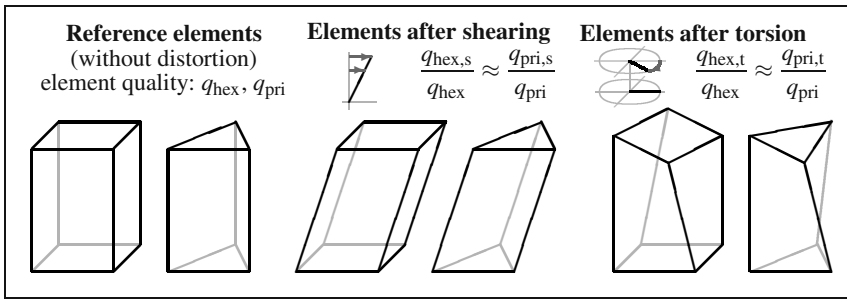
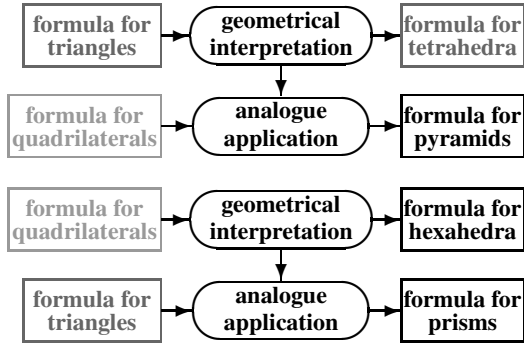


Fig. 9 Explanation of consistency for quality measures of various element types: similar distortions of various element types lead to similar decrease of the quality measure

After all the needed calculation, we get the following formulae for the geometric quality measures of the various element types, equation (6)–(9). Figure 10 provides the notation of edges used in the equations.

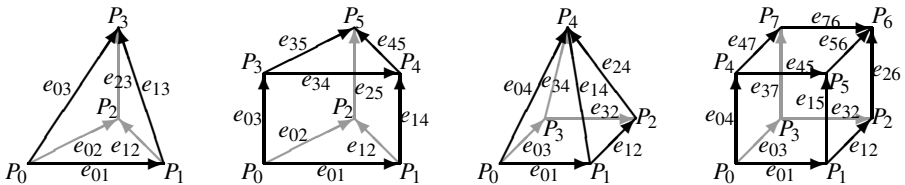


Fig. 10 Notation of the edges of element types used in the formulae of geometric quality

$$Q_{tetra} = \frac{12\sqrt{3} (e_{01} \times e_{02}) \cdot e_{03}}{(e_{01}^2 + e_{02}^2 + e_{12}^2 + e_{03}^2 + e_{13}^2 + e_{23}^2)^{\frac{3}{2}}}, \quad (6)$$

$$Q_{prism} = \frac{3 \cdot 3^{\frac{1}{6}} (V_0 \cdot V_1)^{\frac{1}{6}} (V_{00} \cdot V_{01} \cdot V_{02} \cdot V_{03} \cdot V_{10} \cdot \dots \cdot V_{23})^{\frac{1}{18}}}{2^{\frac{1}{3}} \left((e_{01} + e_{34})^2 + (e_{02} + e_{35})^2 + (e_{12} + e_{45})^2 \right) \cdot |e_{03} + e_{14} + e_{25}|} \quad (7)$$

$$\text{with } V_0 = |e_{01}, e_{02} + e_{12}, e_{03} + e_{14} + e_{25}|, \dots$$

$$\text{and } V_{01} = |e_{01}, e_{02} + e_{12} + e_{35} + e_{45}, e_{03} + e_{14}|, \dots,$$

$$Q_{pyra} = \frac{6\sqrt{3} \left(|e_{01}, e_{03} + e_{12}, E_4| \cdot |e_{32}, e_{03} + e_{12}, E_4| \cdot |e_{01} + e_{32}, e_{03}, E_4| \cdot \dots \right)}{\left(2(e_{01} + e_{32})^2 + 2(e_{03} + e_{12})^2 + E_4^2 \right)^{\frac{3}{2}} \cdot |e_{01} + e_{32}, e_{12}, E_4|^{\frac{1}{4}} \cdot \left((e_{01} + e_{32})^2 + (e_{03} + e_{12})^2 \right) \cdot \dots} \cdot \frac{\dots}{|e_{01} + e_{32}| \cdot |e_{03} + e_{12}|} \quad (8)$$

$$\text{with } E_4 = e_{04} + e_{14} + e_{24} + e_{34},$$

$$Q_{hexa} = \frac{8 \left(|e_{01}, e_{03} + e_{12}, E_2| \cdot |e_{32}, e_{03} + e_{12}, E_2| \cdot |e_{01} + e_{32}, e_{03}, E_2| \cdot \dots \right)}{\dots \frac{|e_{01} + e_{32}, e_{12}, E_2| \cdot |e_{45}, e_{47} + e_{56}, E_2| \cdot \dots \cdot |E_0, e_{03}, e_{04} + e_{37}| \cdot \dots}{|E_0| \cdot |E_1| \cdot |E_2|} \dots \dots} \cdot \frac{\dots \cdot |e_{01}, E_1, e_{04} + e_{15}| \cdot \dots \cdot |e_{32} + e_{76}, E_1, e_{26}|}{\dots} \right)^{\frac{1}{24}} \quad (9)$$

$$\text{with } E_0 = e_{01} + e_{32} + e_{45} + e_{76}, \quad E_1 = e_{03} + e_{12} + e_{47} + e_{56}$$

$$\text{and } E_2 = e_{04} + e_{15} + e_{26} + e_{37}.$$

The incomplete products in equations (7) and (9) go over all the tetrahedra parts of a prism and a hexahedron, respectively. The expression $|x, y, z| = (x \times y) \cdot z$ denotes the triple product which can be calculated as a determinant of the vector entries.

3.2 Modification of Adapted Grids

The idea of grid modification is to avoid bridging refinements for elements of lower geometrical quality. The borders between different refinement levels are moved into elements of a better quality, see Fig. [11](#)

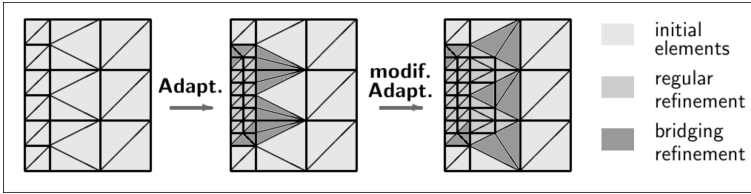


Fig. 11 Idea of the grid modification: the initial elements are of various quality (left), the unmodified adaptation leads to bridging elements of even lower quality (middle), moving the bridging refinements to better initial elements avoids the worst elements (right)

The bridging refinements, the so called green elements of the red-green-method, tend to provide children of a lower quality in comparison with the parents and the corresponding children of a regular refinement. So this method can avoid the worst elements of an adapted grid.

To explain the problems of the method, some algorithmic details are needed. An internal edge refinement trial of the TAU adaptation has three main steps. First as much as possible points are (temporarily) removed. In the second step, the initial refinement is done depending on the error indicators of edges and the target size of the grid. Then, points are added iteratively until all elements have a valid subdivision state as a third step.

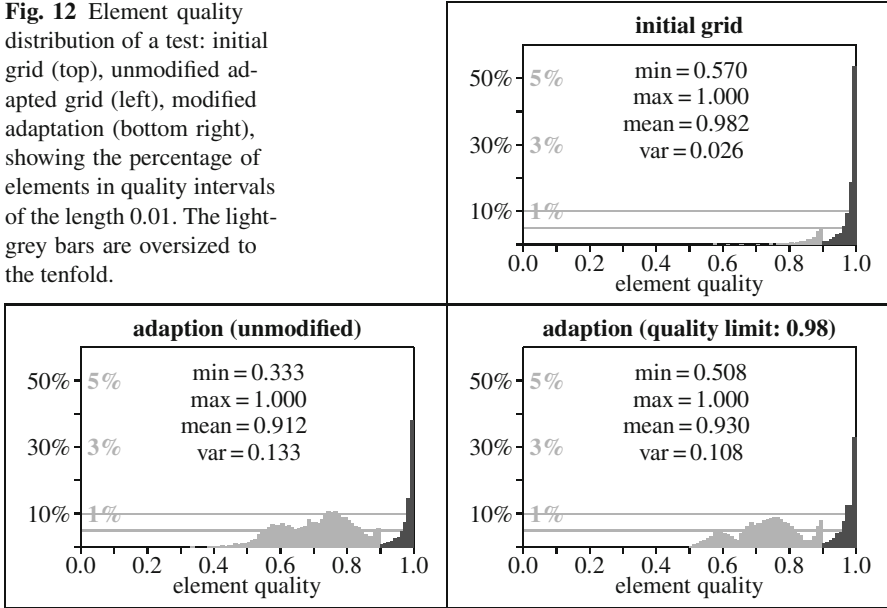
The only place for the modification is the third step, because in the second step the initial borders between refinement levels are determined. Because a bridging refinement can not taken back in this step it has to be re-refined to a regular one. So additional points are needed for the modification. Furthermore, the regular subdivision of a badly shaped element provides more badly shaped child elements than the bridging subdivision. Therefore, regarding the pure numbers, the portion of badly shaped elements could increase though their quality is not worsened in a modified adaptation.

As a first test the triangles of a two-dimensional grid around a RAE-2822 air-foil are adapted without modification and with a forced regular refinement for all elements with a quality lower than a limit of 0.98, see Fig. 12

The quality distributions of the adapted grids have much more elements of lower quality than the initial grid. These are the elements of bridging subdivisions. The modified adaptation can avoid the low quality elements of a quality 0.33–0.5 because they come from bridging subdivisions which become regular ones in the modified case. The portion of high quality elements with quality 0.98–1.0 decreases in the modified adapted grid because a lot of them has to be refined nonregularly, taking the borders of various refinement levels in the modified adapted grid. Because of both effects the behaviour of the average quality of all elements after modification can not be predicted. In this example it is nearly unchanged.

In three-dimensional cases of more complex geometries the element qualities of the initial grids are usually much broader distributed. That is why these effects are overlaying for various qualities. So there are quality slots for which some bridging

Fig. 12 Element quality distribution of a test: initial grid (top), unmodified adapted grid (left), modified adaptation (bottom right), showing the percentage of elements in quality intervals of the length 0.01. The light-grey bars are oversized to the tenfold.



elements are avoided by the modification. But some additional elements from regular subdivisions fall into the same slot to avoid elements of even worse quality.

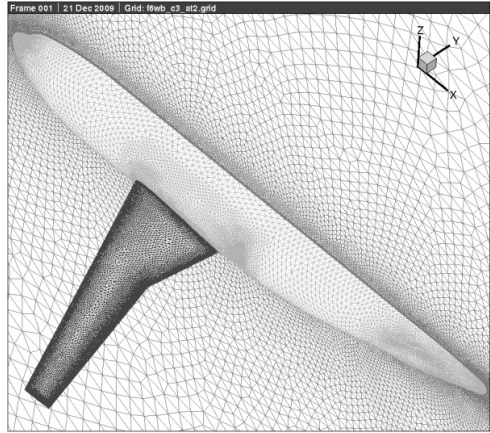
The implemented version of the grid modification requires an input parameter defining the portion of elements which is to be considered for only regular refinement. The TAU adaptation determines the actual quality limit from this portion. In order to have the same behaviour in various adaptation steps, the portion and the limit refer to the initial grid. Furthermore, if an element which is considered for regular refinement has an unrefinable edge, this element is not refined at all.

3.3 Results on Modified Grids

As a more realistic test case, the flow around the DLR-F6 model has been taken. The initial grid was prepared with the Centaur grid generation tool [3], see Fig. 13.

The simulation was done for Mach number of $Ma = 0.75$, an angle of attack of $\alpha = 0.0^\circ$, using the Spalart-Almaras turbulence model in the original version (SAO). After the simulation on the initial grid was fairly converged, the adaptation was performed to increase the point number by 50%, using differences of the solution gradients for refinement indication and using various values of the modification parameter. The modification parameter defines the portion of elements which is considered for only regular refinement. In the test series it was increased from a value of 0, i.e. no modification, to a value of 0.1, the maximum value allowed. Table 3

Fig. 13 Surface of the initial grid of the DLR-F6 model used in the test example for the modified adaptation, containing 2.46 million points, 1.87 million prisms and 8.66 million tetrahedra



shows the different modification parameter values, the actually inserted percentage of new points and how much of these new points are inserted by the edge indicator.

Table 3 List of tested modification parameter values, i. e. the portion of elements prepared for only regular refinement, the actual increase of the adapted grid and the portion of new points which was inserted by the initial edge refinement (and not by the modification)

modification parameter	0.000	0.001	0.002	0.005	0.010	0.020	0.050	0.100
new points in %	49.95	50.02	50.10	50.03	50.31	50.26	50.40	50.47
portion of new points initially inserted in %	100.0	95.26	93.85	90.85	86.66	78.37	49.49	10.60

With an increasing influence of the modification, i. e. with a larger number of elements considered for regular refinement only, the portion of points inserted by indicators decreases, because a lot of points are needed for the regular refinement of badly shaped elements. The table shows that increasing the modification parameter beyond 0.05 seems to be not a reasonable choice in this example, because the ratio of points spent for adaptation of the grid resolution and points spent for the quality manipulation is too small.

The differences of the surface grids for the unmodified case and the case using the modification parameter 0.01 also show this effect, see Fig. 14. In the modified case less points are used to refine the area of the shock system. These points are moved to some badly shaped elements to improve their subdivision. Because these changes mostly are done locally at places where refinement already occurred in the unmodified case, it is hard to find them in the picture. They are located in the areas of the wing-body intersection and behind the trailing edge. In these areas probably the convex or concave geometry forced the grid generation tool to introduce some elements of lower quality.

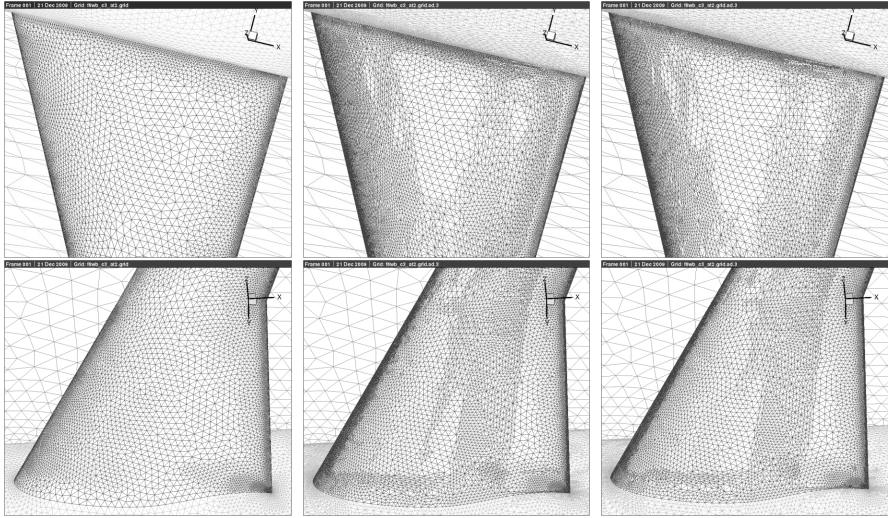


Fig. 14 Influence of modified adaptation on adapted grid: initial grid (left), unmodified adapted grid (middle), adapted grid with modification parameter of 0.01 (right)

The results for the integrated lift and drag coefficients and the residuals after 2000 and 10000 solver iteration steps after the adaptation are listed in the table 4. The results are compared to the corresponding results on a globally refined grid. The globally refined grid includes all the refinements that any local adaptation could do, and the element quality is perfectly preserved because all subdivisions are regular. So the result on the globally refined grid are the best what a local adaptation is able to achieve. The grey-shade of the entries indicates if the result is more accurate or smaller in case of a residual than the corresponding value for the unmodified case. Black entries are better, grey entries are worse than the value of the unmodified case. In this example the best results are achieved with a modification parameter of 0.005–0.010.

Another test example was a simulation on a Solar [8] grid for the HiReTT model. The initial grid has 3.61 million points and consists of 3.16 million hexahedra, 25675 prisms, 81396 pyramids and 2.27 million tetrahedra. The calculation was

Table 4 Residuals and results of various adaptation modifications for the DLR-F6 model

modif. param.	Ref.	0.000	0.001	0.002	0.005	0.010	0.020	0.050	0.100
Res. 12k		6.864e-4	7.452e-4	7.185e-4	7.012e-4	5.244e-4	5.616e-4	6.114e-4	8.682e-4
Res. 20k		4.533e-5	4.466e-5	4.507e-5	4.452e-5	4.689e-5	4.627e-5	4.516e-5	5.302e-5
C_L		4.581e-1	4.613e-1	4.610e-1	4.606e-1	4.594e-1	4.591e-1	4.597e-1	4.663e-1
C_D		2.665e-2	2.730e-2	2.728e-2	2.728e-2	2.727e-2	2.728e-2	2.727e-2	2.723e-2

Table 5 Residuals and results of various adaptation modifications for the HiReTT model

modif. param.	Initial	Global	0.000	0.002	0.005	0.010	0.020
new points in %			102.07	103.28	96.39	100.18	97.28
portion of new p. initially inserted			71.77	70.93	69.31	66.69	57.93
Res. 4000			8.621e-6	8.775e-6	8.816e-6	8.167e-6	6.601e-6
Res. 6000			1.354e-6	1.339e-6	1.320e-6	1.271e-6	1.128e-5
C_L	2.156e-1	2.143e-1	2.146e-1	2.146e-1	2.144e-1	2.145e-1	2.144e-1
C_D	1.024e-2	9.941e-3	9.962e-3	9.962e-3	9.962e-3	9.964e-3	9.971e-3

done for a Mach number of $Ma = 0.85$, an angle of attack of $\alpha = 0.0^\circ$, using the Spalart-Allmaras turbulence model in the original version. After 3000 solver iteration steps the grid was adapted with 100% new points, using the differences of the primitive variables as the refinement indicator. Again, the adaptation was applied using the standard version and various values of the modification parameter.

Table 5 summarizes the results of the adaptation and the following calculation. In this example the best result is achieved for a modification parameter of 0.005. In this case the lift coefficient is approximated a little better, and the deviation of the drag coefficient is not increased.

Some more test calculations for the LANN-wing and the HIRENASD-wing considering the modified adaptation were done. Summarizing the experiences from all tests performed until now, the following remarks can be made:

1. The influence of the described adaptation modification on the results of integral lift and drag coefficients is small compared to the influence of other parameters, e. g. the number of grid points or the turbulence model.
2. The results are almost never worsened for small values of the modification parameter compared to the results using the unmodified adaptation.
3. The results get worse or unpredictable, if the modification parameter approaches the maximum value of 0.1.
4. The influence of the adaptation modification seems to be larger when introducing fewer ($\approx 50\%$) new points than more ($\approx 100\%$) new points in one adaptation step.
5. The residuals tend to fall faster for larger modifications in the first (1000–2000) solver steps after adaptation.
6. If the computation converges well, the residuals for smaller grid modification are often smaller than that for larger grid modification for the converged solution.

The reasons for the unexpected small influence on the results by this adaptation option may be manifold. One possible explanation is that the element shape is less important for the local discretization error than the resolution of the considered examples and grid types. Another interpretation is that the red-green refinement strategy of the TAU adaptation is well suited for the considered grids, i. e. the loss

of element quality in bridging refinements does not affect the solution very much. See Sect. 6 for conclusions regarding the application.

4 Adjoint-Based Error Estimation and Functional Correction

An efficient and consistent way of estimating numerical errors in a functional $I(U)$ of interest, are the so-called dual weighted residual approaches (DWR). Here, one weights the numerical error, represented by the residual $R(U_h)$ of the numerical state solution U_h approximating the exact solution U , by the so-called dual or adjoint state. The idea stems from Johnson, Rannacher et al. [6] in the Finite Element Method (FEM) context. The reason for the weighting of the (local) residual with the adjoint field vector ψ is, because the adjoint is an influence function (i.e. a Green's function) w.r.t. the functional of interest I , for which one has solved the adjoint state equation

$$\frac{\partial I}{\partial U} + \psi^\top \frac{\partial I}{\partial U} = 0. \quad (10)$$

The adjoint field vector has value zero in areas with no impact on the functional $I(U)$, and (very) different from zero in areas of (big) impact. This means, that the adjoint as a weight eliminates local residuals in areas where the error only appears, and increases it in areas where the errors come from.

Unfortunately, there is a difficulty in transferring the DWR to Finite Volume Methods (FVM). The reason is, that one would need the exact adjoint solution for the calculation of the first order error term. Instead, one solves in FEM context the adjoint problem with higher order test and ansatz functions, and this yields then the first order error estimate. But in standard FVM, it is not possible to play around with the order of test and ansatz functions.

One possibility to overcome this difficulty is the repeated extrapolation between certain mesh levels (e.g. from coarse level H to globally refined level h) as suggested by Venditty and Darmofal [13] or Pierce and Giles [10]. This approach has been proved to lead to a good error estimate, given as

$$I_h(U_h^H) - I_h(U_h) \approx (\psi_h^H)^\top R_h(U_h^H), \quad (11)$$

but is not handy e. g. in areas of shocks and w.r.t. memory consumption (due to values needed at globally refined mesh level h).

An alternative idea by Dwight [5] is, to interpret the discretization error $R(U_h)$, at least for the central Jameson-Schmidt-Turkel scheme available in TAU, as dissipation error. This yields instead of

$$I(U_h) - I(U) \approx \psi_h^\top R(U_h) \quad (12)$$

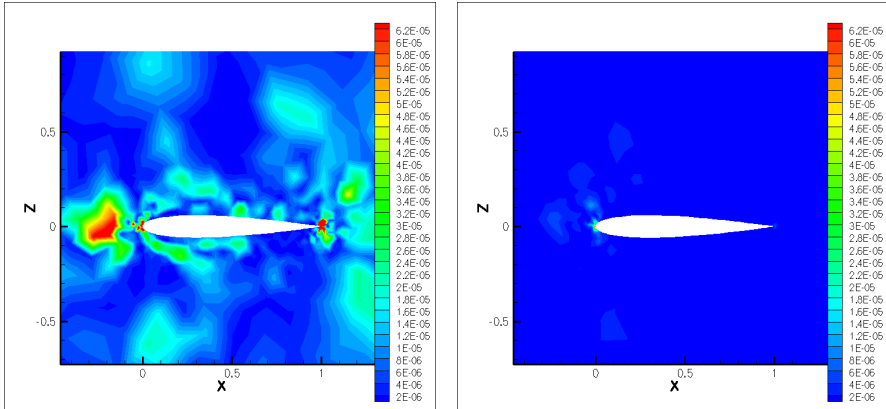


Fig. 15 Adjoint-based error sensor for lift (left) and drag (right)

to the error estimate

$$I(U_h) - I(U) \approx \psi_h^\top \left(k^{(2)} \frac{\partial R}{\partial k^{(2)}} + k^{(4)} \frac{\partial R}{\partial k^{(4)}} \right). \quad (13)$$

Here, $k^{(2)}$ and $k^{(4)}$ are the scaling coefficients for the dissipation of second and forth differences, i. e. first and third order. It turns out that this approach is accurate enough and obviously it is very handy.

This method has been implemented by R. Dwight in the TAU code and has been used by the authors within the project MUNA.

The right hand side of Equation (13), the global error estimate, is to be understood as the integral or sum of the local error estimates

$$\left[\psi_h^\top \left(k^{(2)} \frac{\partial R}{\partial k^{(2)}} + k^{(4)} \frac{\partial R}{\partial k^{(4)}} \right) \right]_{i,j,k} \quad (14)$$

at cell positions $X_{i,j,k}$, and (14) can be used as an indicator or sensor for adjoint-based (and therefore) goal-oriented mesh adaptation. For the realization and application of this local adjoint-based sensor for mesh adaptation in TAU, we refer to Section 5.

In this Section we first present the validation of the local and global adjoint-based error estimate. The difficulty in the validation here is, that one should know about the exact solution to compare with. Therefore, we have chosen an inviscid subsonic NACA0012 flow case and drag as functional of interest. Then we know, that the exact (shock less) solution has zero drag. (Only some spurious drag, i. e. numerical dissipation, remains.)

Figure 15 shows the local sensor (14) for lift and drag. That these sensors, which are the local error estimates, lead to good global error estimates, can be seen in Figure 16.

Furthermore, the adjoint method is used for the efficient quantification of uncertainties in the aerodynamic coefficients caused by variations of the model

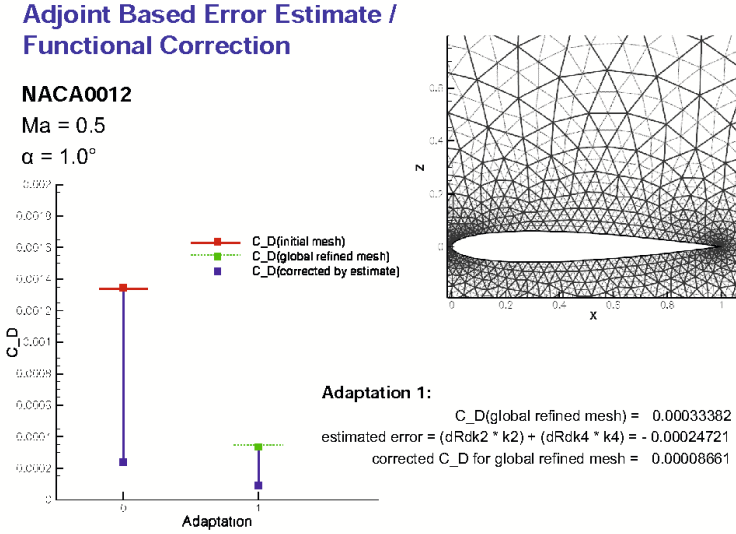


Fig. 16 Adjoint-based global error estimation and functional correction for the drag coefficient on initial and globally refined mesh (NACA0012, $Ma = 0.5$, $\alpha = 1.0^\circ$, inviscid)

parameters of the SAE and the Wilcox- $k-\omega$ turbulence model. Compared to finite differences one is here independent of the number of model parameters w.r.t. the numerical costs. Figures [7] and [8] show a good match of the sensitivities calculated by the adjoint method as well as by finite differences for lift and drag coefficients, caused by variations in the model parameters of SAE and Wilcox- $k-\omega$.

5 Error Indication Based on the Adjoint Solution

This section describes the use of the adjoint information as a sensor for the error indication in the TAU adaptation. The main work was the development and implementation of the adjoint solver which is described in the contribution *Adjoint-Based Error Estimation and Functional Correction* of this volume. The adjoint solver provides a field variable containing a kind of measure for the local discretization error weighted with its influence on the target functional, which may be the integral lift or drag coefficient. This variable serves as an interface to the grid adaptation tool.

The work at the adaptation tool itself which was needed to enable an adjoint-based adaptation was more of technical nature. Because the provided adjoint information has to be regarded as a pointwise error indicator and the adaptation tool works edge orientated, the sum of the sensor variables for the edge points has to be used as the edge indicator. Similar to the differences-based indicators, a weighted

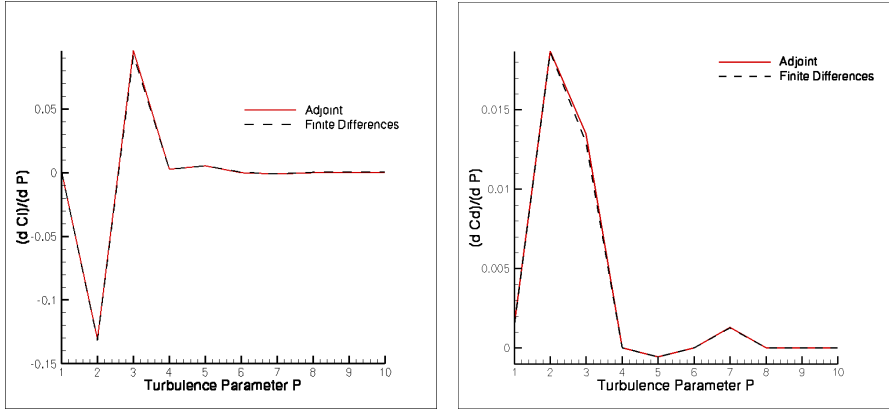


Fig. 17 Sensitivity of lift (left) and drag (right) coefficients w.r.t. variations in the model parameters P of the SAE turbulence model, calculated by adjoint method

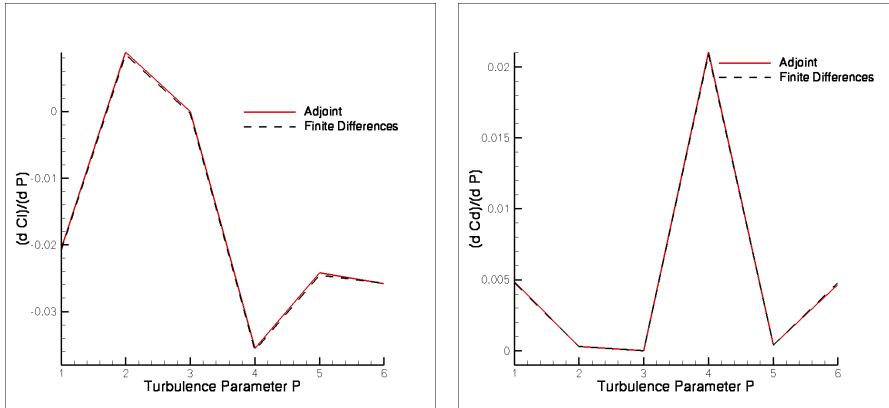


Fig. 18 Sensitivity of lift (left) and drag (right) coefficients w.r.t. variations in the model parameters P of the Wilcox- $k-\omega$ turbulence model, calculated by adjoint method

combination of different variables of the adjoint sensor file can be used for error indication in the TAU adaptation.

The adjoint-based adaptation was tested in a simulation for the HIRENASD-model. The initial grid was generated with the Solar grid generator [8]. It has 3.12 million points and consists of 2.54 million hexhedra, 20758 prisms, 40430 pyramids and 3.19 million tetrahedra. The flow was calculated for a Mach number of $Ma = 0.8$, an angle of attack of $\alpha = 1.0^\circ$ using the Spalart-Allmaras turbulence model in its original version.

After the solution was well converged, the adjoint problem was calculated for the integral drag coefficient as the target functional. Using the adjoint sensors on the one side and the differences of gradients for comparison on the other side, the grid was

Table 6 Results of simulations using the differences-based and the adjoint-based adaptation

		initial grid	adapt +50%	adapt +100%	global +353%
differences-based	C_L	2.783e-1	2.787e-1	2.786e-1	2.789e-1
	C_D	1.181e-2	1.188e-2	1.186e-2	1.171e-2
adjoint-based target: C_D	C_L	2.783e-1	2.773e-1	2.774e-1	2.789e-1
	C_D	1.181e-2	1.175e-2	1.170e-2	1.171e-2

adapted with 50% and 100% new points. Table 6 shows the results of the integral lift and drag coefficients. These results are compared to the results for the initial grid and the globally refined grid. As already discussed in Section 2 and Subsection 3.2, the result on the globally refined grid serves as reference value in case of a locally adapted calculation.

The differences-based adaptation leads to improved results for the lift coefficient, but not for the drag coefficient. It seems that the differences are not appropriate for estimating the local discretization error in this example. Contrary, the adjoint-based adaptation improves the results for the drag coefficient significantly. Because the drag coefficient was used as target functional for the adjoint-based adaptation, an improvement of the lift coefficient result is not expected in this case.

The process was restarted for the example with 50% new points. The results are documented in Table 7 and the resulting grids of the adjoint-based case are shown in Fig. 19. The adaptation used re- and de-refinement and the twice adapted grid has some de-refined grid areas.

The adjoint-based adaptation is significantly more expensive in terms of computational effort. The differences-based adaptation usually can already be performed for a not fully converged solution, especially in case of a computation with multiple adaptations. The resulting grid will not differ very much from that generated by adaptation for the fully converged solution. Different from that, an adjoint calculation requires a very well converged solution. Additionally, the adjoint solution needs computational resources comparable to the original solver and nearly an order of magnitude more memory.

The application of the adjoint-based adaptation requires a careful choice of the control parameters to get the best effect from the much better, but expensive error

Table 7 Results of a twice adapted simulation using the differences-based and the adjoint-based adaptation

		initial grid	adapted +50%	2x adapted +50%
differences-based	C_L	2.783e-1	2.787e-1	2.790e-1
	C_D	1.181e-2	1.188e-2	1.190e-2
adjoint-based target: C_D	C_L	2.783e-1	2.773e-1	2.780e-1
	C_D	1.181e-2	1.175e-2	1.177e-2

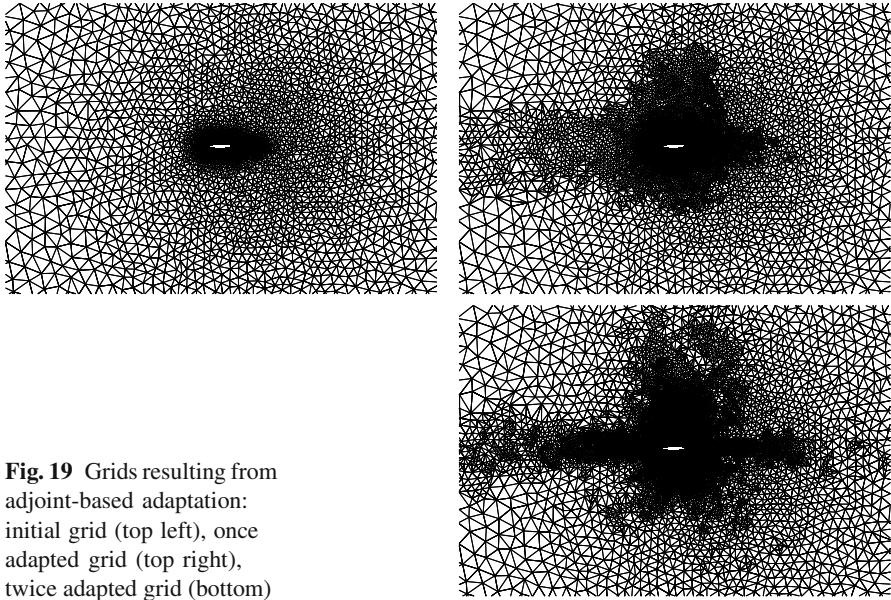


Fig. 19 Grids resulting from adjoint-based adaptation: initial grid (top left), once adapted grid (top right), twice adapted grid (bottom)

estimation. For example, if the percentage of new points is chosen too small, the differences-based adaptation might achieve the same result, using much more points with a similar computational effort. On the other hand, the point number and the resulting grid should not get near the global refinement. In this case the influence of a better refinement indicator would be smaller.

Thus, further systematical tests are needed to find a best practice strategy for the adjoint-based adaptation.

6 Conclusions for the Application

This contribution introduced three development directions of the TAU adaptation tool followed within the framework of the project MUNA, all aiming for improved adapted grids enabling for higher accuracy.

The first one, the investigation and use of the *element decomposability*, see Sect. 2.2 improves the edge refinement algorithm of the TAU adaptation. In effect as much as possible of the grid area is considered for re- or de-refinement, instead of having larger regions which are unintendedly excluded from adaptation. Under the assumption that the refinement indicator provides the correct measure for the necessity to refine edges, this step obviously improves the adapted grids. The better accuracy of the resulting solution was demonstrated for an example, using the adjoint-based error indication.

The second topic, see Sect. 3, provides an option to avoid some of the elements with a low *geometrical quality* introduced by standard adaptation. Because this feature is not as effective as previously thought and slightly increases the computational effort, it is switched off as a default. However, the grid modification seems to be a useful option in some special situations. The idea of avoiding the worst shaped bridging elements in an adapted grid suggests to try this method in configurations with some poorly shaped elements in grid regions affecting the global solution. The stabilizing effect of the grid modification could possibly be used in examples in which the computation converges very slowly after a grid adaptation or a restart is completely impossible. At least the option generates slight grid variations better than inserting random points. So it is an instrument for further investigation of uncertainties caused by grid variations.

The use of the adjoint solution for an *adjoint-based adaptation*, see Fig. 5, significantly improves the accuracy of the result for the target functional obtained on the adapted grids for the investigated test examples. Because of the large effort which is needed for an adjoint-based adapted computation, compared to the conventional differences-based adaptation, a careful cost-benefit analysis has to be done. Some more time and additional tests are needed to find out classes of problems and configurations for which one or the other method is preferable.

References

- [1] Alrutz, T., Orlt, M.: Parallel dynamic grid refinement for industrial applications. In: Proceedings ECCOMAS CFD 2006. Egmond aan Zee, The Netherlands (2006) ISBN 90-9020970-0
- [2] Branets, L., Carey, G.F.: A local cell quality metric and variational grid smoothing algorithm. *Engineering with Computers* 21, 19–28 (2005), doi:10.1007/s00366-005-0309-7
- [3] CentaurSoft: Welcome to CentaurSoft, The solution to grid generation issues for computational engineering problems (2010), <http://www.centaursoft.com>
- [4] Dompierre, J., Vallet, M.-G., Labbe, P., Guibault, F.: An Analysis of Simplex Shape Measures for Anisotropic Meshes. *Computer Methods in Applied Mechanics and Engineering* 194(48-49), 4895–4914 (2005)
- [5] Dwight, R.: Heuristic A Posteriori Estimation of Error due to Dissipation in Finite Volume Schemes and Application to Mesh Adaptation. *Journal of Computational Physics* 227(5), 2845–2863 (2008)
- [6] Johnson, C., Rannacher, R., Boman, M.: Numerics and hydrodynamics stability theory: Towards error control in CFD. *SIAM Journal of Numerical Analysis* 32, 1058–1079 (1995)
- [7] Kallinderis, Y., Kontzialis, C.: A priori mesh quality estimation via direct relation between truncation error and mesh distortion. *Journal of Computational Physics* 228, 881–902 (2009)
- [8] Martineau, D.G., Stokes, S., Munday, S.J., Jackson, A.P., Gribben, B.J.: Anisotropic Hybrid Mesh Generation for Industrial RANS Applications. AIAA 2006-534. AIAA Aerospace Conference, Reno (2006)

- [9] Pebay, P.P.: Planar quadrilateral quality measures. *Engineering with Computers* 20, 157–173 (2004), doi:10.1007/s00366-004-0280-8
- [10] Pierce, N., Giles, M.: Adjoint and defect error bounding and correction for functional estimates. *Journal of Computational Physics* 200, 769–794 (2004)
- [11] Schwamborn, D., Gerhold, T., Heinrich, R.: The DLR TAU-Code: Recent Applications in Research and Industry. In: Wesseling, P., Onate, E., Periaux, J. (eds.) *Proceedings on CD of the European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, The Netherlands (2006)
- [12] Shewchuk, J.R.: What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures. In: *Eleventh International Meshing Roundtable*, Ithaca, New York, pp. 115–126. Sandia National Laboratories (2002)
- [13] Venditti, D., Darmofal, D.: Anisotropic grid adaptation for functional outputs: Application to two-dimensional viscous flows. *Journal of Computational Physics* 187, 22–46 (2003)

Application of Mesh Modifications and Adjoint Error Estimates

S. Albensoeder

Abstract. Two methods for mesh modification are considered to improve hybrid meshes for CFD calculations. The first method is an adaptation with new sensors. The new sensors are based on an adjoint approach to calculate the sensitivity with respect to a goal function. Here the sensitivity of lift, drag and pitching moment was calculated with respect to the numerical dissipation terms. The second method is a local mesh modification of the unstructured part of the hybrid mesh based on an algebraic quality measure. For an a posteriori improvement the flow properties can be included to build a new anisotropic metric. Both new methods were applied to industrial relevant test cases.

1 Introduction

One problem of today computational fluid dynamics (CFD) is the discretization of the computational domain. Due to the limits of computational resources the discretization of the domain is not fine enough. Therefore the discretization can have a significant effect to the results.

A common approach to reduce this uncertainty is the adaptive refinement of the grid where errors occur. In the past several sensors (e.g. gradient based, reconstruction based) were developed to detect these underresolved regions. A sensor which computes the sensitivity of a discretization with respect to a specified goal function was introduced by [13]. The sensor was computed by solving an adjoint problem. One bottleneck of the method was that the final sensor was computed on the isotropic refined mesh instead of the original mesh. For complex configurations with a high number of grid points the demands to the computational resources are very high. In this investigation the sensors of [2] were used. This method computes the

S. Albensoeder

P3 Voith Aerospace GmbH, Flughafenallee 26, 28199 Bremen, Germany

e-mail: Stefan.Albensoeder@p3voith.com

sensitivity with respect to numerical dissipation terms. By this ansatz the error estimation can be done without any mesh refinement step.

Another approach is the improvement of a given mesh by local modifications. This improvement can be related to improve badly shaped elements and to orientate elements in the direction of the flow.

The uncertainty due to influences of the mesh generation drives the limitation that small influences can only be computed on the same or slightly modified mesh. One example is the deformation of the geometry due to aerodynamic loads. To reduce the uncertainty the whole mesh will be deformed to avoid a new meshing. Unfortunately this deformation can cause inverted elements which foreclose a new CFD computation. These cells have to be repaired which can also be done by the introduced local mesh modification.

In the next section the investigated methods are described. In section 3 the methods were applied to industrially relevant test cases. Finally a conclusion and an outlook are given.

2 Methods

For this study two different methods were used. The first approach solves an adjoint problem to get an error estimate of a functional. This error estimate is used as sensor for adaptation. The second method is based on the local modification of the unstructured part of the mesh to increase the quality of the mesh. All computations were performed with the TAU solver from the DLR (Deutsches Zentrum für Luft- und Raumfahrt).

2.1 Adjoint Error Estimation Method

The adjoint error estimation method uses a solution of an adjoint problem as sensor for an adaptation. The sensor is goal-oriented which means that for a specific goal function (e.g. C_L , C_D or C_{My}) the sensitivity on the error of this goal-function is locally computed. Based on this error estimate the mesh is refined to improve the results with respect to the specified goal function.

The original sensor of [13] is based on an estimate for the goal function I on a globally refined mesh

$$I_h(U_h^H) - I_h(U_h) \approx (\psi_h^H)^T R(U_h^H) \quad (1)$$

by the adjoint ψ times the residuum R of the flow U , where the subscript h means results on the isotropically refined mesh, the superscript H denotes an extrapolation from the coarse to the fine mesh. So on the right hand side of (1) the adjoint ψ is computed on the original mesh and extrapolated to the fine mesh. The flow quantity

U is extrapolated to the fine mesh. After that the residuum is calculated on the fine mesh from this extrapolated vector.

To avoid the extrapolation and the calculation of the residuum on the fine mesh a new sensor was developed by [2]. The idea of [2] is to assume that the major part of the discretization error comes from the dissipation error. The error estimate is then given by

$$I(U_h) - I(U) \approx \psi^T \left(k^{(2)} \frac{\partial R}{\partial k^{(2)}} + k^{(4)} \frac{\partial R}{\partial k^{(4)}} \right). \quad (2)$$

The corrected value of the goal function is

$$I(U_h) \approx I(U) + \psi^T \left(k^{(2)} \frac{\partial R}{\partial k^{(2)}} + k^{(4)} \frac{\partial R}{\partial k^{(4)}} \right). \quad (3)$$

The related sensor for the adaptation is the absolute value of the local product

$$s_{i,j} = \left| \psi_{i,j}^T \left(k_{i,j}^{(2)} \frac{\partial R_{i,j}}{\partial k^{(2)}} + k_{i,j}^{(4)} \frac{\partial R_{i,j}}{\partial k^{(4)}} \right) \right|. \quad (4)$$

2.2 Mesh Manipulation

For optimizing a mesh, a quality measure for its elements has to be defined. Here the used quality measure based on the so-called *mean ratio*

$$q_i = \begin{cases} 4\sqrt{3} \cdot \frac{A_i}{\sum_{j=1}^3 l_{ij}^2} & \text{for 2D meshes (triangles)} \\ 12 \cdot \text{sign}(V_i) \cdot \frac{\sqrt[3]{(3V_i)^2}}{\sum_{j=1}^6 l_{ij}^2} & \text{for 3D meshes (tetrahedron)} \end{cases}, \quad (5)$$

where A_i is the area, V_i the volume and l_{ij} the edge lengths of the element i (see figure 1a). Basically the measure is a ratio between the volume (3D) or area (2D) and the edge lengths.

For three-dimensional cases this quality measure was extended to pyramids by splitting the pyramid into four tetrahedra by introducing a mid point on the basis of the pyramid (see figure 1b).

The implementation of the mean measure allows to use local anisotropic metrics. The modified metric can be helpful if more information about the flow e. g. a preliminary solution is available. In this case the orientation of the elements to the local flow is considered. Due to this new metric the edge lengths, area and volume are measured in the space of the new metric \mathcal{M} . The size functions in the new metric are then given by

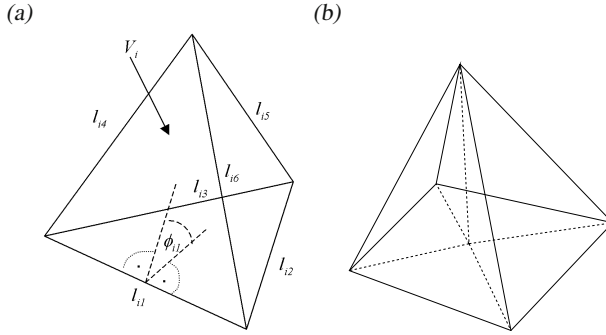


Fig. 1 (a) Nomenclature of variables on a tetrahedral element i . (b) Splitting of a pyramid into four tetrahedrons.

$$l_{ij}^{\mathcal{M}} = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \cdot \mathcal{M} \cdot (\mathbf{x}_i - \mathbf{x}_j)}, \quad (6)$$

$$A_i^{\mathcal{M}} = \sqrt{\det(\mathcal{M})} \cdot A_i \quad (\text{for 2D meshes}), \quad (7)$$

$$V_i^{\mathcal{M}} = \sqrt{\det(\mathcal{M})} \cdot V_i \quad (\text{for 3D meshes}). \quad (8)$$

For this study the metric was derived from the Hessian of the local Mach number Ma_{local}

$$\mathcal{H} = \frac{\partial}{\partial x_i \partial x_j} \text{Ma}_{local}. \quad (9)$$

To get a positive definite metric the Hessian was decomposed to its eigenvalues.

$$\mathcal{H} = \mathbf{R} \cdot \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \cdot \mathbf{R}^T. \quad (10)$$

The new local metric is then defined by the absolute values of the eigenvalues

$$\mathcal{M} = \mathbf{R} \cdot \begin{pmatrix} |\lambda_1| & 0 & 0 \\ 0 & |\lambda_2| & 0 \\ 0 & 0 & |\lambda_3| \end{pmatrix} \cdot \mathbf{R}^T \quad (11)$$

To improve the quality of a three-dimensional mesh, four different methods are implemented to modify the unstructured:

- edge swapping for up to 8 surrounding tetrahedrons [6, 10]
- face swapping [face to edge swap in 10]
- edge collapsing [9]
- combined smoothing [11, 1, 4] with an optimizer for not continuously differentiable goal functions [5, 6, 3]

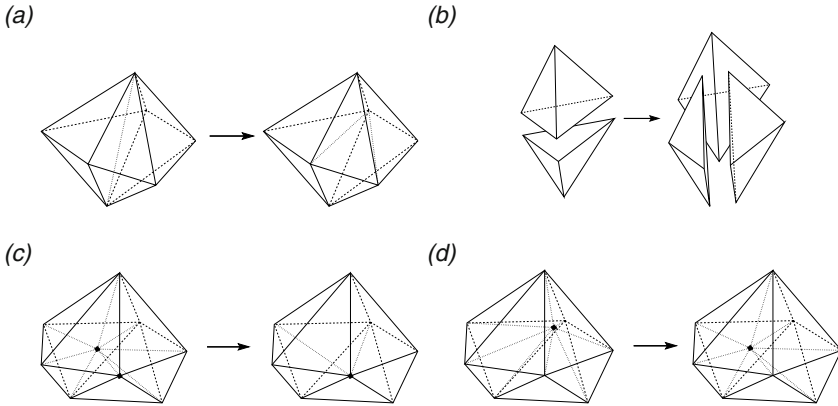


Fig. 2 Examples for (a) edge swapping, (b) face swapping, (c) edge collapsing and (d) combined smoothing

Each method modifies the mesh locally and acts on tetrahedrons. The combined smoothing and the edge collapsing were extended to allow modifications on points connected with pyramids. An example for each method is sketched in figure 2(a)–(d). For two-dimensional grids only edge swapping and the movement of nodes are implemented.

For pointwise optimization due to movement of a node, a goal function has to be defined which combines the quality of surrounding elements S_i of a node i . Here the minimal quality

$$\bar{g}(S_i) = \min_{j \in S_i} q_j \tag{12}$$

was used as goal function for the optimizer.

A modification of the grid is tried if the geometrical constraints allow a modification. Additionally the following demands have to be always fulfilled to accept a modification step:

- the minimal quality is larger than zero (to avoid inverted elements)
- the minimal quality is larger than the global minimal quality
- the goal function has to be improved (in the case of node movement)
- improve the mean quality (and therefore the global quality of the mesh)

3 Results

3.1 Adaptation by an Adjoint Error Estimate

The adjoint error estimate as sensor for adaptation is tested on two configurations. The first configuration is a clean wing/body configuration of the DLR-F6 geometry

[8, 12]. The second configuration is the high-lift wing/body configuration TC 217 with deployed flaps and slats. For both configurations the numerical results are compared to wind tunnel measurements.

3.1.1 DLR-F6

This test case was defined for the Third AIAA CFD Drag Prediction Workshop [see 12]. The flow parameters for the DLR-F6 wing/body configuration are $Ma \approx 0.75$, $Re \approx 5 \cdot 10^6$ and $T_{ref} = 322.22$. The meshes were taken from the workshop¹.

The first test on this configuration is a comparison of a mesh refinement study and an adaptation series with the new sensors. For the refinement study a computation on a coarse ($N_p = 2464385$), a medium ($N_p = 5102446$) and a fine ($N_p = 8535263$) mesh was performed. For all computations the lift is targeted to $C_L = 0.5$ and the Spalart–Allmaras turbulence model was used. Outgoing from the coarse mesh the adaptation was repeated five times with the new sensor given in equation (4). The number of points increases within each adaptation step by 30%. Additional to the flow computation in each adaptation step the adjoint error estimation was computed.

The values of the angle of attack α , the drag C_D and the pitching moment C_{M_y} of the refinement study are plotted in figure 3(a)–(c) as function of $N_p^{-2/3}$. The crosses show the results of the coarse, medium and fine mesh. The circles and squares connected by solid lines are showing the results of the adaptation with the new sensors for lift and drag, respectively. Signs connected by dotted lines denote values which are corrected by the adjoint error estimate (3). The results from the refined meshes show an ambiguous behaviour regarding the convergence for all plotted coefficients. In contrast the results from the adaptation series converge approximately to distinct values. The corrections by the adjoint error estimate decrease continuously so that finally the corrected values converge to the uncorrected values. The values of the finest meshes of the adaptation series are higher than the values from the finest mesh, e. g. $\Delta C_D \approx 10DC$.

A comparison with the results given in [12] shows that the results of the refined and of the adapted meshes are lying in the range of other codes. Additionally [12] make a statistical analysis of the values for drag. Their estimate for the mean of the drag is $C_D = 0.0269$, the standard deviation is $\sigma = 0.0006$. Here the values of the adaptation are significantly out of this range.

The experimental results are taken from [7]. The measured values are shown in figure 3(a)–(c) by a dashed line. For the angle of attack and the pitching moment the adapted grids are lying closer to these experimental results than the results of the refined meshes. For the drag the values on the refined meshes are closer to the experiment.

In the second test a polar is computed. Again the flow conditions are $Ma \approx 0.75$, $Re \approx 5 \cdot 10^6$ and $T_{ref} = 322.22^\circ\text{K}$. Instead of a fixed target lift coefficient here the angle of attack was varied in the range of $\alpha = [-5^\circ, 2^\circ]$. The polar is computed on

¹ See <http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw>

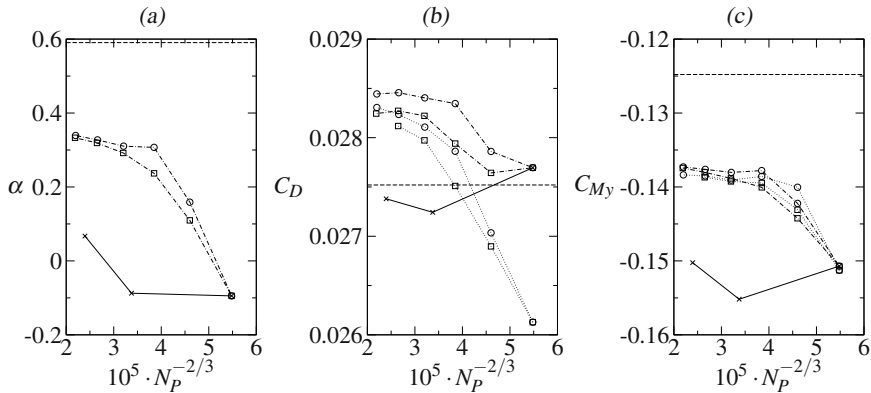


Fig. 3 Angle of attack α (a), drag C_D (b) and the pitching moment C_{M_y} (c) as function of the grid points N_P for the DLR-F6 model. The lift is kept constant to $C_L = 0.5$. \times denotes the result on the base meshes. \circ and \square connected by a dash-dotted line marks the results of the adaptation with an adjoint sensor with a sensitivity to lift and drag, respectively. The results connected with the dotted lines are corrected by the adjoint error estimate. The dashed line denotes the experimental results taken from [7].

the coarse, medium and fine mesh. To avoid the computation of an adjoint solution for each angle of attack, four adapted meshes from the previous adaptation series for lift and drag are used. Here the mesh from the first (*ad1*) and the third adaptation (*ad3*) step with $N_P \approx 3.2 \cdot 10^6$ and $N_P \approx 5.5 \cdot 10^6$ are chosen, respectively.

In figure 4(a) the lift is plotted as function of the angle of attack α . The results on the coarse, medium and fine mesh are denoted by \circ , $+$ and $*$. On the coarse and medium mesh the results are similar. On the fine mesh the slope is lower than on the coarser meshes. The computed lift on the adapted meshes *ad1* for lift and drag are nearly identical within the line thickness. In comparison to the fine mesh the lift is shifted by $\delta C_L \approx 1LC$. The computed lift on the meshes *ad3* show the same behaviour. Only the shift to the fine mesh is $\delta C_L \approx 2LC$.

The lift as function of the drag is plotted in figure 4(b). On the refined mesh series the drag reduces by the mesh refinement. The strongest variation of the lift as function of the drag is observable in the region of minimal drag where the curve of the fine mesh is shifted to the left by $\delta C_D \approx 10DC$ in comparison to the drag computed on the coarse mesh. For large angles of attack the reduced drag is nearly completely compensated by the reduced values of the lift so that the curves of the refinement series are close together. For the adapted meshes the shift at the minimal drag is only $\delta C_D \approx 3 - 5DC$ with respect to the results on the coarse mesh. In contrast to the refined meshes the deviation for large angles of attack are getting higher by the number of points.

In figure 5 the pitching moment is plotted as function of the lift. The results for the refined mesh series are ambiguous and no trend is observable. The pitching moment on the adapted meshes *ad1* and *ad3* increases by each refinement.

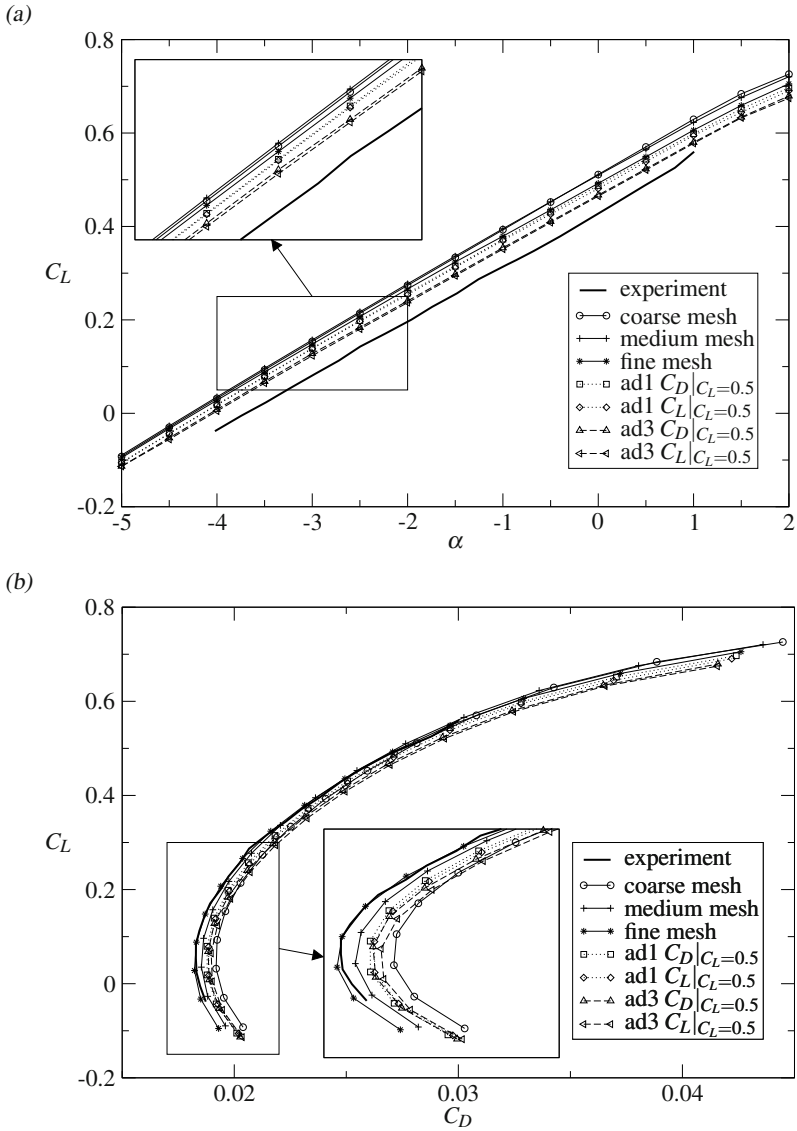


Fig. 4 (a) Lift C_L as function of the angle of attack α and (b) polar of C_L as function of C_D for the DLR-F6 model. \circ , $+$ and $*$ denotes the results on the coarse, medium and fine mesh. \square and \diamond marks the results on the first adapted mesh at $C_L = 0.5$ with respect to the lift and drag, respectively. \triangle and \triangleleft denotes the results computed on the meshes of the third adaptation step of figure 3. The solid line denotes the experimental results taken from [7].

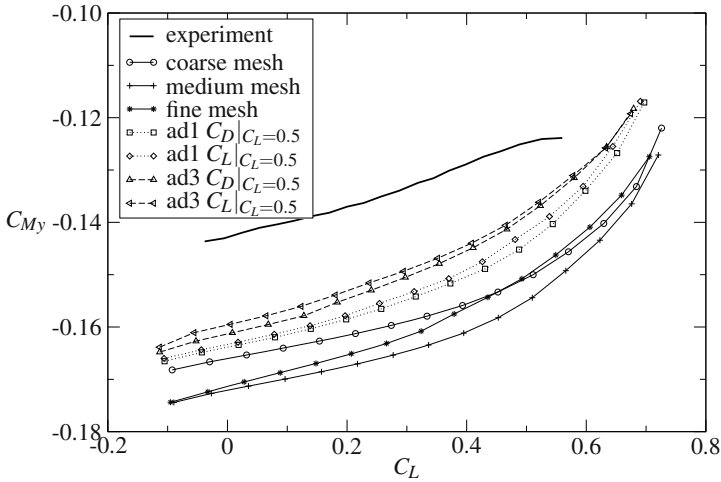


Fig. 5 Pitching moment C_{My} as function of the lift C_L for the DLR-F6 model. The labels are identical to the labels of figure 4.

The measurements of [7] are plotted in the figures 4 and 5 as a solid line. Like in the convergence study by trend the lift (angle of attack) and pitching moment of the adapted meshes fits more to the experimental results than the coefficients computed on the refined meshes. The measured lift as function of the drag fits more to the refined meshes.

3.1.2 TC 217

The adaptation sensor based on an adjoint error estimate is also applied to the second test case of the TC 217 high-lift configuration. The model was a wing/body configuration with deployed slat and flaps. The geometry was previously used in the EURO-LIFT project. The flow parameters are $Ma \approx 0.18$, $Re \approx 1.5 \cdot 10^7$ and $T_{ref} \approx 114^\circ K$. All computations are performed with the Spalart–Allmaras turbulence model.

For this configuration the polar was computed on a mesh with $N_P = 10733766$ grid points. Outgoing from this mesh for several angles of attack an adaptation with the new sensor (4) was performed. Like for the DLR-F6 model the adaptation was repeated several times. The number of points increases within each adaptation step by 30%. In contrast to the DLR-F6 test case convergence problems occur on this configuration, e.g. the convergence was too slow and the computational effort too high or the adjoint computation diverges. However, for most of the angles of attack one adaptation iteration was successfully performed. In figure 6 the results of this first test were plotted. The + shows the result of the computation on the base mesh. The corresponding experimental values were plotted as solid line. The major differences are observable in the linear region and near the maximal value of lift $C_{L,max}$.

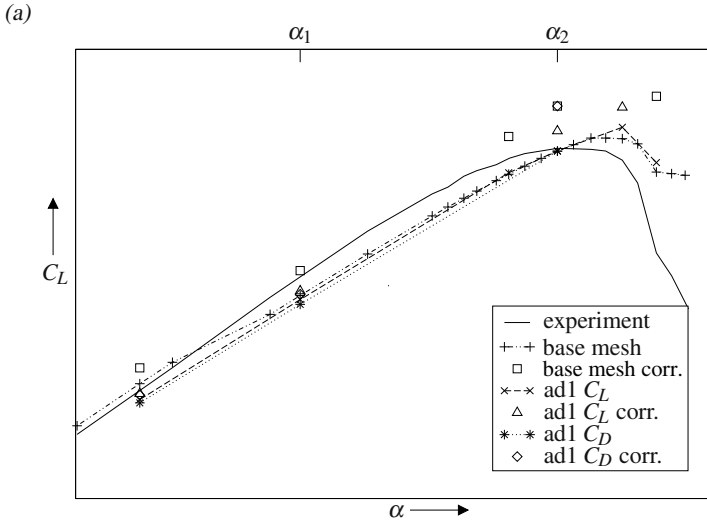


Fig. 6 Lift C_L as function of the angle of attack α for the TC-217 model. + and o denote the results on the base mesh and the corrected values by the adjoint error estimation, respectively. \times and * mark the results of the first adaptation step with a sensor based on lift and drag, respectively. The Δ and \diamond are the corrected values. The solid line denotes the experimental results.

For lower angles of attack the computational results have a step-like shape. On the other side of the curve the angle of attack and the corresponding lift is too high. The values of the lift corrected by the adjoint error estimate (\square in figure 6) are higher than the original ones.

Outgoing from these results the mesh was adapted for several angles of attack α with the adjoint error estimate for lift (\times in figure 6) and drag (* in figure 6), respectively. For both sensors an improvement of the linear region is observable. The step like behaviour vanishes. In the nonlinear region the results (if they are available) are similar to the results on the base mesh. The corrected values for the adapted grids are higher than the original values. Unfortunately most of the adjoint computations fail. Due to this experiment and the high computational effort four adapted meshes are selected from the linear and the nonlinear region. The four selected meshes are the meshes adapted by the lift or drag sensor for $\alpha = \alpha_1$ and $\alpha = \alpha_2$ (see figure 6), respectively. The results are plotted in figure 7. The results on the base mesh are marked by +. The results of the lift adapted mesh at α_1 and α_2 are denoted by o and \square , respectively. The results of the drag adapted meshes are denoted by Δ and \times .

In figure 7(a) the lift is plotted as function of the angle of attack. In the linear region the values on the adapted meshes are close together. The deviation between the meshes $ad1 C_{L,\alpha_1}$ and $ad1 C_{D,\alpha_1}$ is $\delta C_L \approx 1LC$. The offset between the meshes $ad1 C_{L,\alpha_2}$ and $ad1 C_{D,\alpha_2}$ is of the same order. The offset between the adapted meshes and the base mesh is about $\delta C_L \approx 10LC$ for $\alpha \lesssim \alpha_1$. The step like behaviour in the lift curve at $\alpha \approx \alpha_1$ vanishes due to the adaptation. In the nonlinear region

close to maximal lift the results of the adapted meshes diverge from each other. The results on the meshes which are adapted at α_2 show a higher maximal lift coefficient ($\delta C_L \approx 3LC$ with respect to the base mesh) and also the angle of attack of maximal lift is shifted by $\delta\alpha \approx 1^\circ$ relative to the results on the base mesh. The maximal lift value on the meshes adapted in the linear region at α_1 have a lower value than the lift on the base mesh. The offset in the position of the maximal lift is $\delta\alpha \approx 0.5^\circ$.

In figure 7(b) the lift is plotted as function of the drag. Again the results on the adapted meshes are close together in the linear region. At maximum lift the results on the meshes adapted at α_1 diverge from the values of the meshes adapted at α_2 .

All computations differ from the experimental results. The slope of $C_L(\alpha)$ is lower for the computational results than in the experiment. The values of maximal lift are higher than in the experiment. The same holds for $C_L(C_D)$ in figure 7(b). However, in the linear range an improvement by the new method is observable due to the vanishing of the step like behaviour in the linear region. The improvements for the maximal lift are ambiguous.

3.2 Mesh Manipulation

3.2.1 Repairing of Meshes

Negative elements occur many times by applying a deformation on the mesh, e. g. for CFD/CSM coupling. The appearance of negative cells permits a CFD computation on those grids. To avoid a new meshing of the deformed geometry and to measure effects below the uncertainties of mesh effects the negative cells have to be repaired. Here the repairing of negative elements was successfully demonstrated on two configurations. An isotropic metric was chosen to calculate the quality measure.

The first test case was a clean wing/body configuration with horizontal tail planes. To trim the configuration the horizontal tail plane was deformed. Due to deformation of the tail plane 83 elements with a negative volume occur (figure 8). These inverted elements are tetrahedrons and pyramids. By applying the mesh modification all negative elements vanish.

The second test case was a commercial aircraft including wing, body, nacelle, pylon, vertical tail plane and flap track fairings in a high-lift configuration (figure 9a). Due to the applied deformation, 26 tetrahedrons have a negative volume. These elements are located in the slot between the flap inboards and the wing (figure 9b). Again, by applying the implemented mesh modification all negative tetrahedrons are inverted to valid elements with a positive volume.

In many cases the generation of negative elements can also be prevented by applying the mesh modification on elements with a low quality before the mesh is deformed. For example in the previous case of trimming the horizontal tail plane, negative elements were prevented if the mesh was improved before the deformation step is performed.

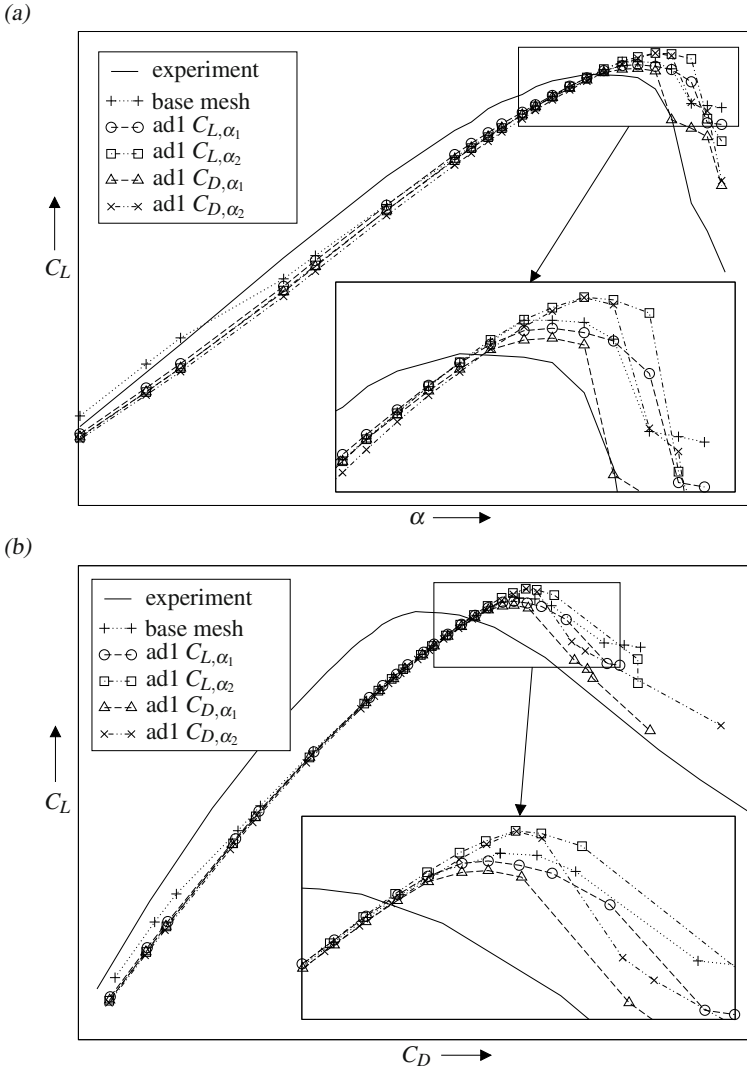


Fig. 7 (a) Lift C_L as function of the angle of attack α for the TC-217 model. (b) Polar for C_L and C_D of the TC-217 model. + marks the results computed on the base mesh. \circ and \square denote the results of the meshes adapted with the adjoint lift sensor for α_1 and α_2 . \triangle and \times show the results for the adapted grids with respect to the drag. The solid line shows the experimental results.

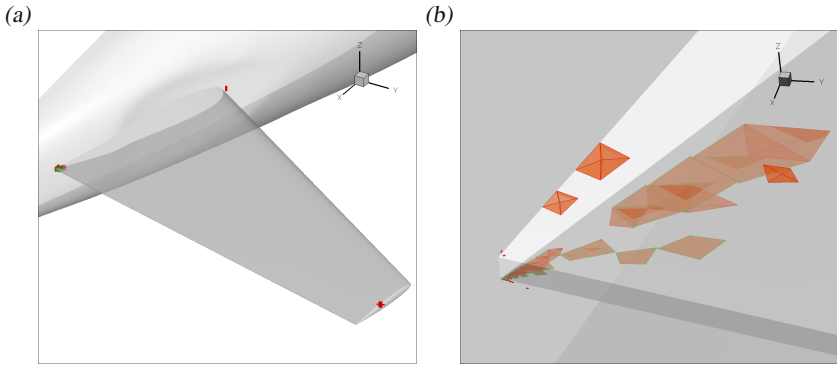


Fig. 8 (a) Negative pyramids and tetrahedrons near by a deformed tail plane. (b) Detail of (a).

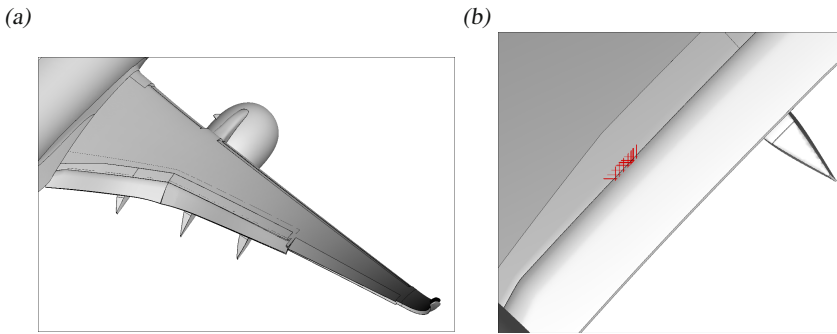


Fig. 9 (a) Deformed wing in a high-lift configuration. Shown is the original wing and the deformed wing. Darker regions have a higher deformation. (b) Detail of (a) with shown negative elements marked by crosses.

Another application of the mesh modification is the improvement of the quality of the unstructured part of the mesh when the computation on the mesh diverges. In figure 10(a) an example is shown where many tetrahedrons have a bad shape. By applying the mesh modifications the number of badly shaped elements decreases significantly (figure 10(b)). The number of tetrahedrons with a dihedral angle less than one degree decreases from $N_{\xi < 1^\circ} = 6840$ to $N_{\xi < 1^\circ} = 29$. The number of tetrahedrons with a dihedral angle less than five degree decreases from $N_{\xi < 5^\circ} = 34598$ to $N_{\xi < 5^\circ} = 252$. Note that after the improvement of the mesh a converged CFD solution could be achieved.

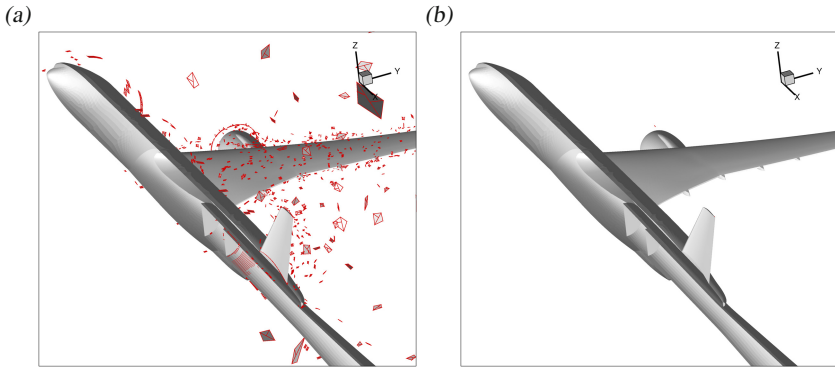


Fig. 10 Mesh with many badly shaped tetrahedral elements before (a) and after (b) the mesh modification. In both figures elements with a dihedral angle less than one degree are shown.

3.2.2 Mesh Modification with Anisotropic Quality Measures

The mesh modification was applied to the DLR-F6 test case described in section 3.1.1 to see the influences on the aerodynamic coefficients. Again a target lift computation for $C_L = 0.5$ was performed. After the flow computation the coarse, medium and fine mesh was modified by using a quality measure which uses an anisotropic metric. The metric was derived from the Hessian of the local Mach number. The flow was recomputed on the anisotropic mesh. This procedure was performed twice.

In figures 11(a)–(c) the angle of attack, the drag and the pitching moment are plotted as function of the grid points N_P . The results computed on the base meshes are denoted by \times . The results of the first and second mesh modification step is marked by \circ and \square . The major improvements are observable for the drag on the coarse mesh, but the differences to the original mesh are relatively small in comparison to the changes caused by an adaptation with the new sensors. The major changes are observable in the resolution of the wake. In figure 12 the eddy viscosity is shown on four different meshes. The eddy viscosity on the coarse and fine base meshes (figure 12a,c) dissipates in the unstructured part very quickly, where on the anisotropically modified meshes (figure 12b,d) the wake is more resolved behind the wing, even on the coarse mesh. The improved wake resolution is observable on the whole configuration.

In a second test the mesh modification with an anisotropic quality measure was applied to the TC-217 configuration (see section 3.1.2). For the test the mesh modification with the anisotropic quality measure was applied twice to the base mesh for $\alpha = \alpha_1$. Again the metric was derived from the Hessian of the local Mach number. The results are presented in figure 13. The $+$ and \square denote the results on the original and the modified mesh, respectively. Due to the mesh modification the lift decreases

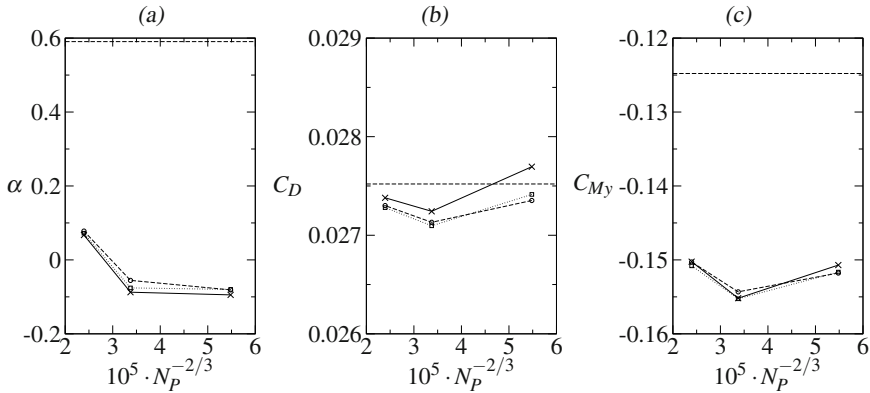


Fig. 11 Angle of attack α (a), drag C_D (b) and pitching moment C_{M_y} (c) as function of the grid points N_p for the DLR-F6 model. The lift is kept constant to $C_L = 0.5$. \times denotes the result on the base meshes. \circ and \square mark the results on the meshes modified once and twice by the modification with an anisotropic quality measure, respectively. The dashed line denotes the experimental results taken from [7].

by $\delta C_L \approx 8LC$. The drag increases by $\delta C_D \approx 100DC$. The result on the modified mesh increases the differences to the experimental results.

4 Conclusion

The methods of adjoint error estimates and local mesh modification are applied to several test cases. The main focus was set on the global aerodynamic coefficients. Only for the mesh repairing the focus was set on the validity and the computability of meshes.

For the adaptation series of the DLR-F6 with the adjoint error estimates no final conclusion can be given. The numerical values given by [12] are widely spread. The corresponding experimental data show for some coefficients improvements and for some not. At least one should note that a comparison of the experimental results for $Re = 3 \cdot 10^6$ measured in the NASA NTF and in the ONERA S2MA facility shows as well differences especially in the drag [7]. If one projected these differences to the measurements at $Re = 5 \cdot 10^6$ the adapted meshes would show an improvement for all coefficients.

However, the results of the adaptation series converge for both used sensors to similar values which are different from the values of the finest mesh. The results on the meshes which are locally modified to satisfy a quality measure based on an anisotropic metric are close to the results computed on the original meshes.

For the high-lift configuration the adjoint approach improves the results in the linear region of the polar. The values of maximal lift are ambiguous and depend on the used meshes. The deviations of the results on the meshes which are optimized to

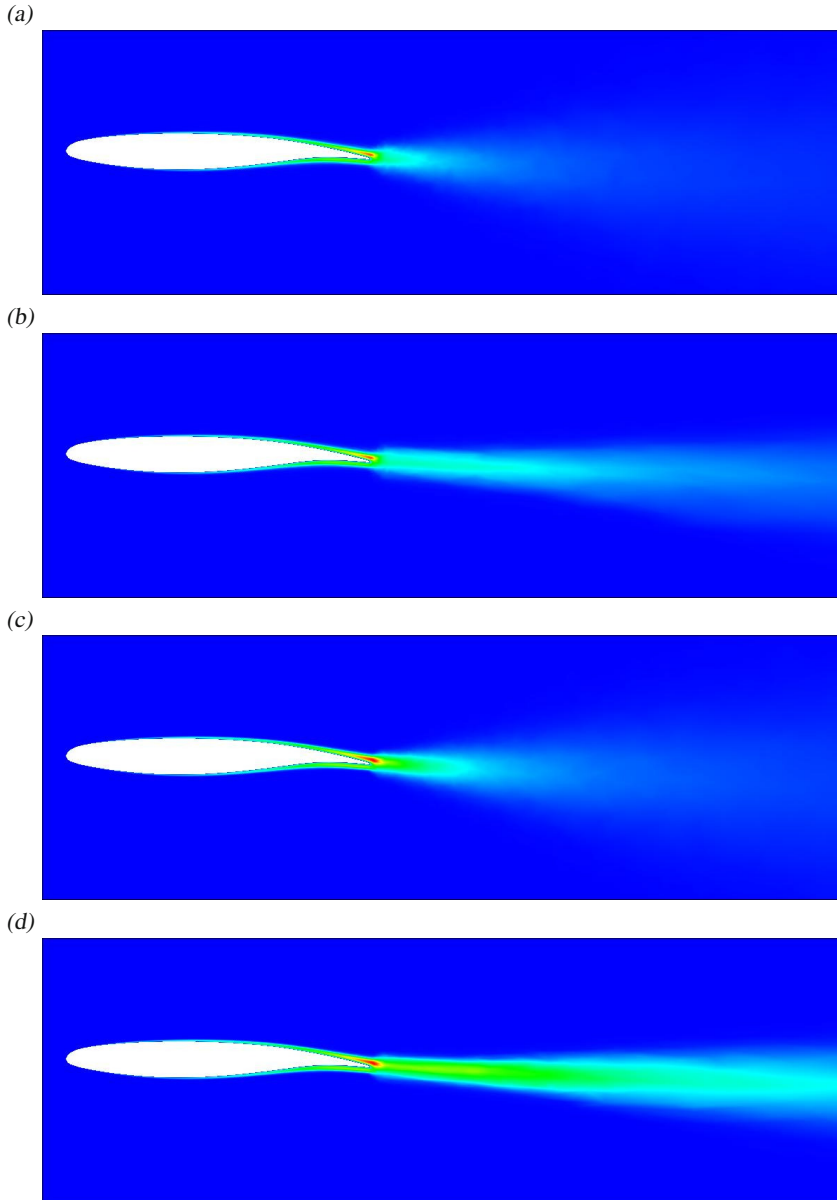


Fig. 12 Eddy viscosity of the wake at $\eta = 0.514$ for (a) the base coarse mesh, (b) the anisotropic coarse mesh, (c) the base fine mesh and (d) the anisotropic fine mesh of the DLR-F6 configuration

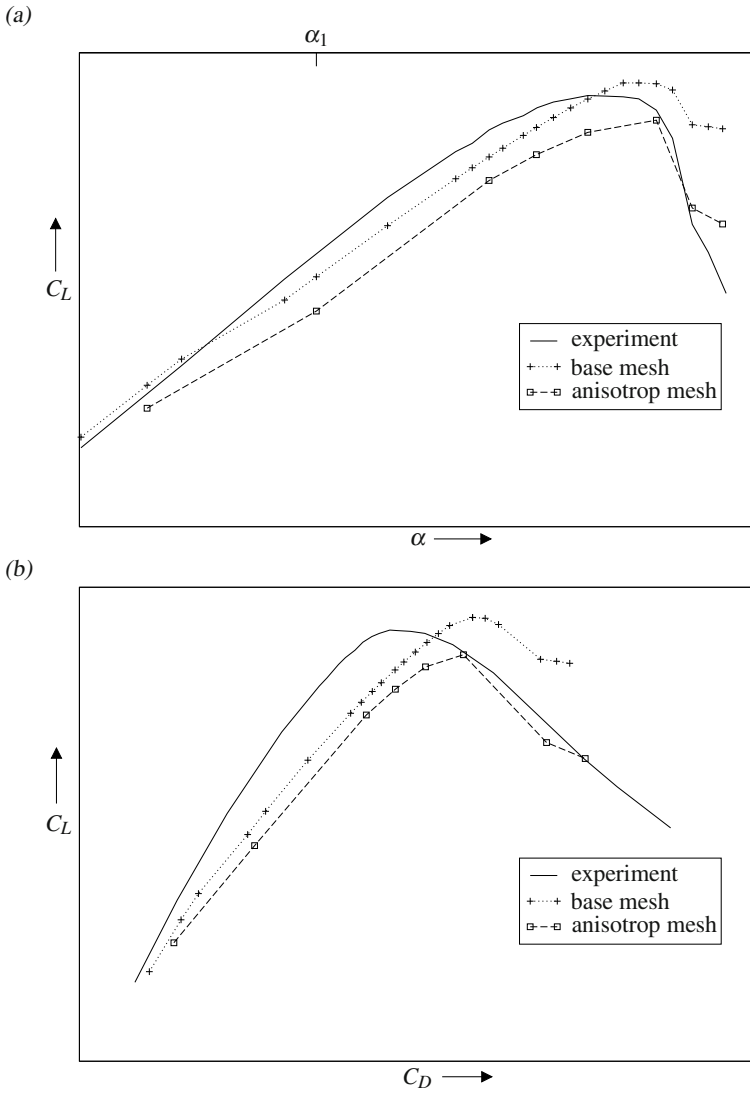


Fig. 13 (a) Lift C_L as function of the angle of attack α for the TC-217 model. (b) Polar for C_L and C_D of the TC-217 model. + marks the results computed on the base mesh. \square denotes the results on the modified meshes. The solid line shows the experimental results.

an anisotropic quality measure increase. The reason for the strong deviations have to be analyzed.

The results for the adjoint approach show additionally that the strategy to use one grid for the whole polar works even if the adaptation was performed only for a specific angle of attack. This strategy reduces significantly the computational effort and makes the adjoint approach applicable.

The repairing of meshes was successfully demonstrated for several meshes. This method is useful if a computation on a mesh due to numerical errors or negative elements fails and a new mesh generation is not desirable.

5 Outlook

[13] couples the adjoint adaptation with the feature of anisotropy. He demonstrated that with this mixed approach the convergence of the coefficients is much faster than with an isotropic mesh adaptation. [13] states that the adjoint approach balances the inaccuracies which can occur by the anisotropic meshes and uses the positive effects of anisotropic meshes. This has to be tested with the present methods.

Additionally the introduced error estimation approach of [2] tries to reduce the error with respect to numerical dissipation terms. The original work of [13] tries to reduce the error with respect to the discretization itself. A comparison of both approaches would also be interesting.

Acknowledgements. The author is grateful to R. Heinrich and E. Elsholz for providing the shown test cases for the mesh repairing.

References

- [1] Canann, S.A., Tristano, J.R., Staten, M.L.: An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. In: 7th International Meshing Roundtable, Dearborn, Michigan, pp. 479–494. Sandia National Labs (1998)
- [2] Dwight, R.: Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *J. Comp. Phys.* 227, 2845–2863 (2008)
- [3] Freitag, L., Jones, M., Plassmann, P.: A parallel algorithm for mesh smoothing. *SIAM J. Sci. Comput.* 20, 2023–2040 (1999)
- [4] Freitag, L.A.: On combining Laplacian and optimization-based mesh smoothing techniques. *ASME* 220, 37–43 (1997)
- [5] Freitag, L.A., Ollivier-Gooch, C.: A comparison of tetrahedral mesh improvement techniques. In: Proceedings of the Fifth International Meshing Roundtable, pp. 87–100. Sandia National Laboratories, Albuquerque (1996)
- [6] Freitag, L.A., Ollivier-Gooch, C.: Tetrahedral mesh improvement using swapping and smoothing. *Int. J. Numer. Meth. Engrg.* 40, 3979–4002 (1997)

- [7] Gatlin, G.M., Rivers, M.B., Goodliff, S.L., Rudnik, R., Sitzmann, M.: Experimental investigation of the DLR-F6 transport configuration in the National Transonic Facility. In: 26th AIAA Applied Aerodynamics Conference, Honolulu. American Institute of Aeronautics and Astronautics, Hawaii, AIAA 2008-6917 (2008)
- [8] Laflin, K.R., Klausmeyer, S.M., Zickuhr, T., Vassberg, J.C., Wahls, R.A., Morrison, J.H., Brodersen, O.P., Rakowitz, M.E., Tinoco, E.N., Godard, J.L.: Data summary from Second AIAA Computational Fluid Dynamics Drag Prediction Workshop. *J. Aircraft* 42, 1165–1178 (2005)
- [9] Li, X., Shephard, M.S., Beall, M.W.: 3D anisotropic mesh adaptation by mesh modification. *Comput. Methods in Appl. Mech. Engrg.* 194, 4915–4950 (2005)
- [10] Pain, C.C., Umpleby, A.P., de Oliveira, C.R.E., Goddard, A.J.H.: Tetrahedral mesh optimization and adaptivity for steady-state and transient finite element calculations. *Comput. Methods in Appl. Mech. Engrg.* 190, 3771–3796 (2001)
- [11] Shephard, M.S., Georges, M.K.: Automatic three-dimensional mesh generation by the finite octree technique. *Int. J. Numer. Meth. Engrg.* 32, 709–749 (1991)
- [12] Vassberg, J.C., Tinoco, E.T., Mani, M., Brodersen, O.P., Einfeld, B., Wahls, R.A., Morrison, J.H., Zickuhr, T., Laflin, R., Mavriplis, D.J.: Summary of the third AIAA CFD Drag Prediction Workshop. In: 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada. American Institute of Aeronautics and Astronautics, AIAA 2007-260 (2007)
- [13] Venditti, D.A.: Grid adaptation for functional outputs of compressible flow simulations. PhD thesis, Massachusetts Institute of Technology, Boston, USA (2002)

Part II

Turbulence Modeling

Minimization and Quantification of Errors and Uncertainties in RANS Modeling

Tobias Schmidt, Charles Mockett, and Frank Thiele

1 Introduction

Owing to their affordable computational cost relative to higher-fidelity approaches such as large-eddy simulation (LES), statistical turbulence models are currently the principle workhorse for the simulation of turbulent flows in industrial aerodynamics. However, significant problems arise from the inherent empiricism of such Reynolds-averaged Navier–Stokes (RANS) approaches. First and foremost, the current state of the art is that no universally-applicable model is available. Instead, a very large number of different RANS models exist, with varying degrees of mathematical complexity and with calibration valid for limited classes of flows. This state of affairs is further compounded by the experience that more complex formulations do not necessarily deliver better results. For these reasons, the choice of turbulence model for an engineering simulation has a strong impact on the quality of the results obtained. In addition to this, simulation results using a fixed RANS model show a strong sensitivity to other aspects of the simulation setup, most notably the grid. For external aerodynamics applications, the spatial resolution of the thin boundary layer regions is seen to be particularly important. All these factors lead to a very high dependency on the decisions made by the engineer in setting up the simulation, and strong reliance is placed on a combination of best practice guidelines (BPG) and user experience.

The motivation of this work is therefore the development of a series of extensions to the TAU flow solver, intended to reduce this user burden and to improve the quality of simulation results in an industrial environment. The approach taken is the development of a range of sensors to check important grid design parameters

Tobias Schmidt · Charles Mockett · Frank Thiele
TU Berlin, Müller-Breslau-Str. 8, D-10623 Berlin
e-mail: {tobias.schmidt, charles.mockett}@cfcd.tu-berlin.de
frank.thiele@cfcd.tu-berlin.de

and to detect the occurrence of flow phenomena known to be correlated with high model-dependency. For the grid error sensors, this amounts to the incorporation of BPG within the software itself. With these sensors in place, the next step is to attempt to quantify the error introduced. For this, the feasibility of an empirical approach is assessed, whereby the solution sensitivity to various error mechanisms is established on simple datum test cases and extrapolated to more complex flows. The implemented module provides the engineer with enhanced textual and graphical feedback, drawing attention to possible problems and suggesting appropriate steps to improve results and minimise such errors.

2 Classification of Errors and Uncertainties

The structuring of this work is based upon the distinction between errors and uncertainties, for which the definitions given in the BPG for industrial CFD published by ERCOFTAC [11] are adopted.

Errors

The term error is used to refer to problems arising from mis-use of the flow solver or turbulence model and is distinct from the functionality of the applied model. Errors hence lead to different predictions for the same flow using the same model and a prominent example is the use of an insufficiently fine grid. Errors can in theory be avoided through proper solution setup (i.e. by an experienced user and/or through adherence to an appropriate set of BPG). Nonetheless, the problem of errors should not be under-estimated since they are often hard to identify and to avoid, particularly in complex industrial configurations. Furthermore, advanced knowledge of the flow solution is required to correctly generate the grid. Examples include the choice of the skin friction normalised wall-normal cell size at solid boundaries, y^+ , the wall-normal expansion ratio of the grid inside the boundary layer, the recommendation of wall-normal grid cells and the capturing of the complete boundary layer with prismatic cells.

Uncertainties

In contrast to errors, the term uncertainty is used to refer to loss of predictive accuracy that occurs due to lack of knowledge about the true flow physics. In the scope of this work, this implies shortcomings of the RANS models in describing the turbulence. Deviation of an error-free solution with “reality” can hence arise

due to uncertainties. The varying reliability of different turbulence models for different flow phenomena, e.g. separating/reattaching flows and shock-boundary layer interaction, is primarily considered. Compared to errors, BPG concerning RANS modelling uncertainties are more vague in nature and less accessible to a “Boolean” type of treatment: Simple statements along the lines of “Model A is best for flow type X” have remained elusive.

3 Development of the Sensors

In order to quantify and handle the errors and uncertainties arising, sensors for their detection were developed and implemented. These sensors scan the boundary layer and flow field to ascertain the level of adherence to relevant best practice guidelines. These sensors are developed and calibrated on the basis of a generic test case using a reference solution with minimal error. A sensitivity study with isolated mechanisms was carried out first of all, following which the interaction of errors was investigated for mixed errors. The results were parameterized to give simple expressions for the error magnitude as a function of the relevant sensor output. The applicability of these derived functions to more complex test cases is tested.

3.1 Development of the Error Sensors

At the beginning of the investigation the grid convergence study was performed on the flat plate test case ($Re = 2.1e6$, $Ma = 0.1$, $u_0 = 33m/s$) with a fully structured grid. For this the turbulence model of Spalart–Allmaras with Edward–modification (SAE) was used, which is the specified standard model in TAU. The grid expansion ratio near the wall and the wall distance y_0 of the wall nearest grid point were varied until there were no significant changes in the solution. The hereby identified grid was used as a reference grid and as a basis for the further investigations. For all investigations eight different turbulence models were applied — two one–equations models (SAE and Spalart–Allmaras (SAO)), three two–equation models (Wilcox $k-\omega$, Menter–SST und LEA $k-\omega$) and three EARSM models (Rung RQEVm, Wallin & Johansson 2D Mean Flows (WJ2D) und Hellsten EARSM $k-\omega$).

3.1.1 Orthogonality in the Boundary Layer

To investigate the influence of the inclination of the grid in the boundary layer the reference grid ($y_0 = 1e-6$, $y^+ = 0.08$, $ratio = 1.1$) were inclined to achieve angle variations of $0deg$ to $70deg$ in $10deg$ steps. Angles against the flow direction were tested only between $-10deg$ and $-30deg$. Here the results show the same influence

as the angles in the flow direction. Hence the independence of the angle direction is assumed.

The variation of the inclination in the boundary layer shows with the one-equation models (SAO and SAE) a massive influence in the results (fig. 2 & 3). Compared with the results of the reference grid the friction coefficients differ by more than 100% at an inclination of 60deg (fig. 5). The same behavior is shown in the drag coefficient c_D . This influence is however the result of inaccuracies arising from the computation of the wall-normal distance d (fig. 4), which is reflected most strongly in the results of the models with the strongest dependency on d . A lower error occurs with the models SST and Hellsten EARSM and the error is negligible for the remaining models. With increasing inclination the calculated wall distance differs from the correct values. This leads to the demonstrated errors with all turbulence models that depend on the wall distance. For this simple geometry, the analytically correct value of d could easily be determined. Replacing the computed d with the correct value gave results with negligible sensitivity to the grid inclination.

Implementation of the Inclination Sensor

For an automatic evaluation by the flow solver of the boundary layer thickness, the grid stretching ratio and the skewness near the wall, the boundary layer had to be scanned at every grid point on the surface. For this, the neighboring point furthest from the wall was detected, starting at each surface point and working in the wall normal direction. In this manner the entire boundary layer is traversed until the total pressure reaches 99.9% of the free stream total pressure [2]. The skewness of the grid is defined as the average deviation from the wall normal vector and the stretching ratio as the maximum stretching ratio found inside the boundary layer. These quantities are saved at the respective surface points. From the results of the test computations for the isolated error mechanisms the magnitude of the expected error was determined. The friction coefficient at the positions $x_1 \dots x_4$ was computed

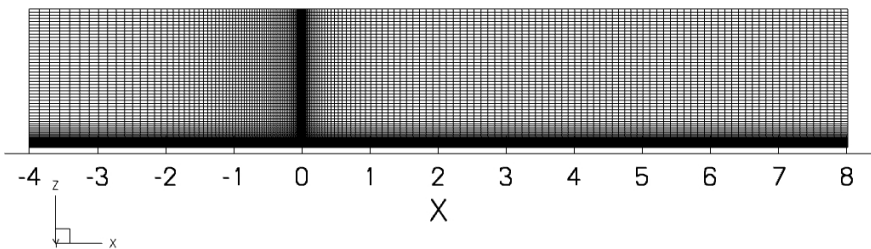


Fig. 1 Structured grid of the flat plate test-case with 52,000 grid points

and compared by normalization with the results of the reference grid. The error factor of c_f is therefore given by

$$c_{fn} = \frac{c_f - c_{f,ref}}{c_{f,ref}} = \frac{c_f}{c_{f,ref}} - 1 \tag{1}$$

and the percentage error as $c_{fn} \cdot 100\%$. This was carried out for all four stream-wise positions to verify independence of this variable. For the prediction of the errors in c_f the skewness is assumed as constant. The error factor c_{fn} was parameterized with a 3rd order polynomial ($p(x) = a_0 + a_1x + a_2x^2 + a_3x^3$) and the coefficients a_i were identified for all investigated models.

3.1.2 Wall Distance y^+

Again the reference grid was the basis for the investigations of the normalized wall distance y^+ . For this the wall nearest grid point was varied between $y^+ = 0.07$ to $y^+ = 7$.

In the results (Fig. 6) a high sensitivity to the wall distance y^+ is seen, with a widening of the boundary layer and modification of τ_w , and hence u_τ . Depending on the model, at $y^+ \approx 1$ these values achieve differences up to 2.5% compared to the reference computations. With further variations the differences exceed 25% quite fast. As the figure 7 shows, the behavior of the error in c_f is independent of the position in the boundary layer. $y^+ \geq 7$ leads to divergence of the simulations or

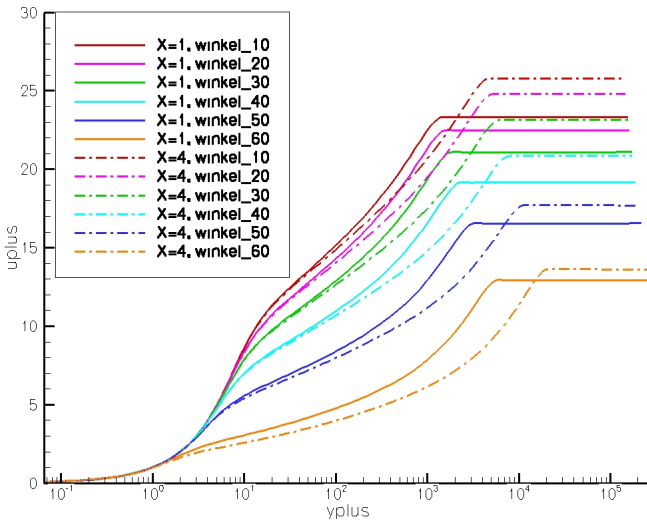


Fig. 2 Flat plate — SAE model — graph of u^+ / y^+ with variations of grid inclination in the boundary layer at $x = 1.021$ and $x = 4.124$

unexpected results. The results shown all apply the default low-Re wall treatment and the use of wall functions were not examined.

Implementation of the y^+ Sensor

For the implementation of the y^+ -sensor the behavior of the error was parameterized with a polynomial 3rd order for each turbulence model. The polynomial represents the difference to the results with the reference grid in %. Because of the low-Re boundary condition and the lack of error data for $y^+ \geq 7$, the difference is set to 100% if y^+ exceeds the viscous sub-layer.

3.1.3 Maximum Grid Expansion Ratio in the Boundary Layer Resolution

For the investigation of the near wall grid expansion the reference grid is again used as a basis. The expansion ratio was varied between $r = 1.1$ up to $r = 2.0$. Again all eight turbulence models were applied.

In the results with small variations the influence of the expansion ratio could be seen clearly. The behavior of the error prediction with different turbulence models is similar (Fig. 8). With a bigger ratio up to $r = 2.0$, the differences to the reference solution lead to 8% in the friction coefficients c_f in some cases. Greater values of the expansion ratio weren't investigated. To ensure that the error prediction doesn't depend on the local streamwise position within the boundary layer, the normalized

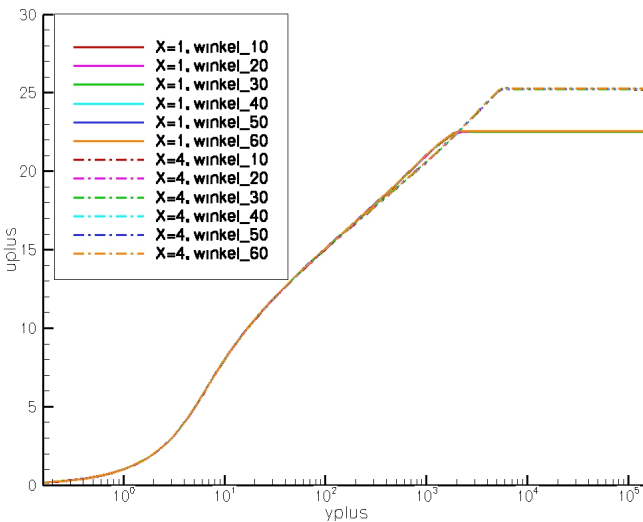


Fig. 3 Flat plate — WCX model — graph of u^+/y^+ with variations of grid inclination in the boundary layer at $x = 1.021$ and $x = 4.124$

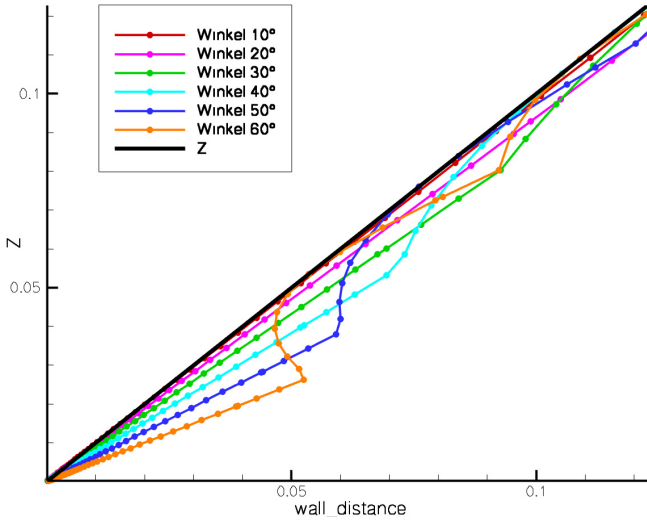


Fig. 4 Computation of the wall distance in TAU with variations of the grid inclination in the boundary layer in comparison to the correct wall distance z

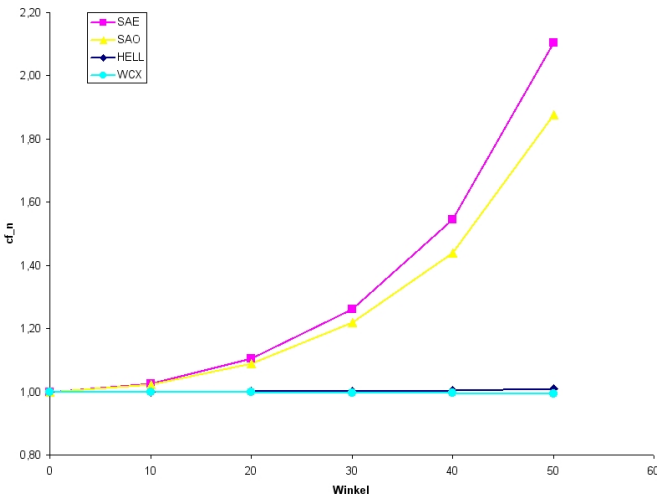


Fig. 5 Flat plate — influence of the grid inclination on the prediction of the friction coefficient c_f , normalized with c_f of the reference grid with SAE, SAO, Hellsten EARSM and Wilcox $k-\omega$ model

error factor of friction c_{fn} has been evaluated at several positions. The behaviors of c_{fn} over r as well as y^+ were compared between these locations. As the comparison of the c_{fn} progression reveals, the error predictions are nearly identical at all

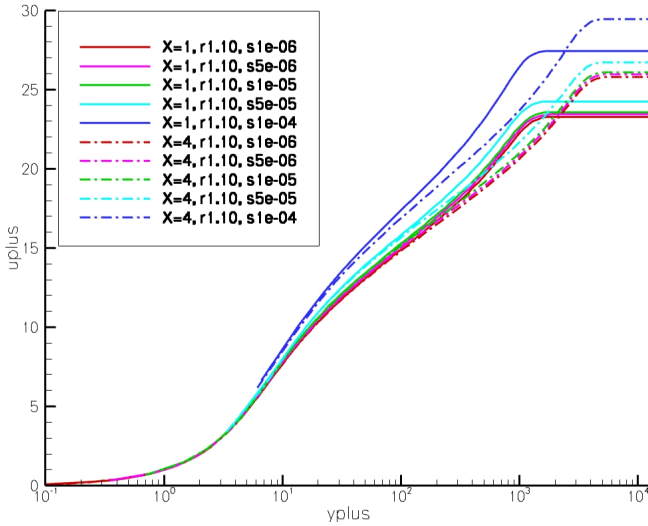


Fig. 6 Flat plate — Hellsten EARSM model — graph of u^+/y^+ with variations of y^+ at $x = 1.021$ and $x = 4.124$

positions. A minimal deviation occurs only at high levels of y^+ and r . According to that c_{fn} can be assumed independent of the stream-wise location in boundary layer.

Implementation of the Expansion Sensor

For the implementation of the sensor for the expansion ratio again the boundary layer has to be scanned. The identified maximal ratio is assumed as constant at the related surface grid point. The behavior of the error was parameterized with a polynomial 3rd order for each turbulence model. This polynomial again represents the difference to the results with the reference grid in %. The difference is set to 100% if the expansion ratio leaves the investigated range of values.

3.2 Development of the Uncertainty Sensors

As already mentioned, the uncertainties considered are mostly model-related and can be the result of the specification or weaknesses of the model itself. Reasons for uncertainties could be a lack of information of the physical background of the investigated flow problem.

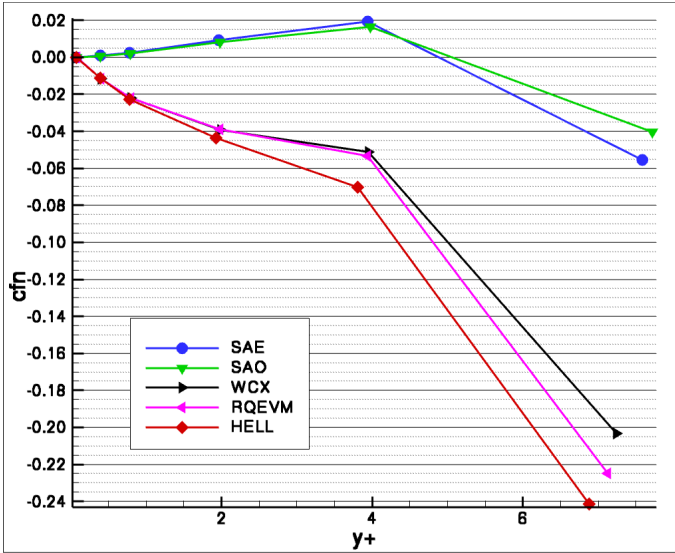


Fig. 7 Flat plate — influence of the wall distance y^+ on the prediction of the friction coefficient c_f , normalized with c_f of the reference grid with SAE, SAO, Hellsten EARSM, RQEVm and Wilcox $k-\omega$ model

3.2.1 Boundary Layer

For the investigation of the model dependencies on boundary layer development again the flat plate test-case was applied. The grid is identical as the reference grid for the error analysis.

The variation of the turbulence model leads to visible differences in the friction coefficients and boundary layer thickness (fig. 9). Here the results and the experimental data were compared at given positions x . In comparison to the experiment the simulations show differences in the fully developed boundary layer at $x = 4.124m$ of up to 26.4% in the boundary layer thickness δ (WCX) and 6.7% in the friction coefficients c_f . The error range of the different models is $\pm 12.71\%$ in δ and $\pm 3.58\%$ based on an average value over all models.

Implementation of the Boundary Layer Sensor

For the implementation of this sensor again the boundary layer has to be scanned to identify their thickness and the number of wall normal grid points. As the investigations show the choice of the turbulence model itself is an uncertainty. Depending on the model the boundary layer thickness or the friction is differing to each other. The greatest problem is the difference in the gradients of the various values especially the friction coefficient. The reason for this is the calibration of the models and their

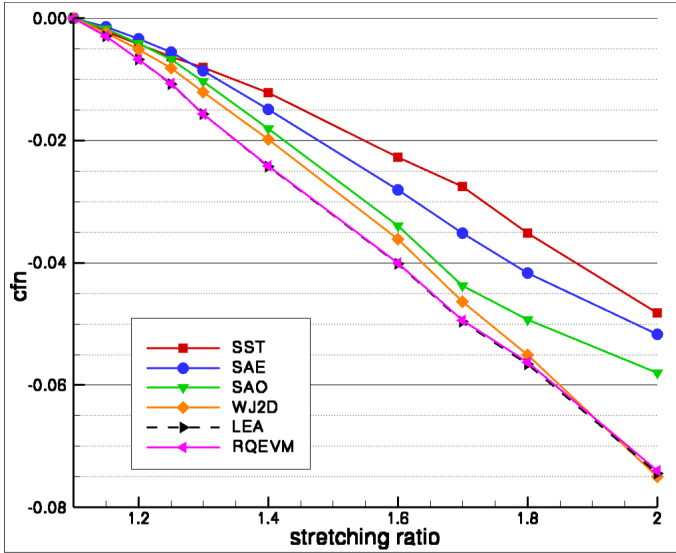


Fig. 8 Flat plate — influence of the grid expansion ratio on the prediction of the friction coefficient c_f , normalized with c_f of the reference grid with Menter SST, SAE, SAO, Wallin & Johansson EARSM, RQEVM and LEA model

different development of the boundary layer. Hence it is impossible to give a correct prediction for a local error. Despite that the sensor gives feedback to the user about the boundary layer thickness and the applied grid resolution.

3.2.2 Separation

The quality of the different turbulence models to predict the correct position of a separation is very variable. This has a great influence on the correct prediction of the drag and lift coefficients. For the investigation the 2D test-case Onera-A at 13.3 angle of attack was chosen. The test-case has a separation at 89.5% chord-length on the suction side with $Re = 2.1e6$ and $Ma = 0.15$. The grid consists of 530,000 grid points (fig. 10) and was already tested on grid convergence in the ECARP-project. The order of convergence of the turbulence models differs slightly, but the results should also be comparable.

The variation of the turbulence models leads to visible variations in the separation position and also to different levels in c_p and c_f (fig. 11 & 12). Compared to the experiment the deviations are 8.9% in the lift and up to 73.62% in the drag coefficient (SST). This strong deviation in the drag coefficient is in part due to the fully-turbulent computations applied: A user-specified transition location is known to ignore results for this test case. The range of variation in the forces reaches

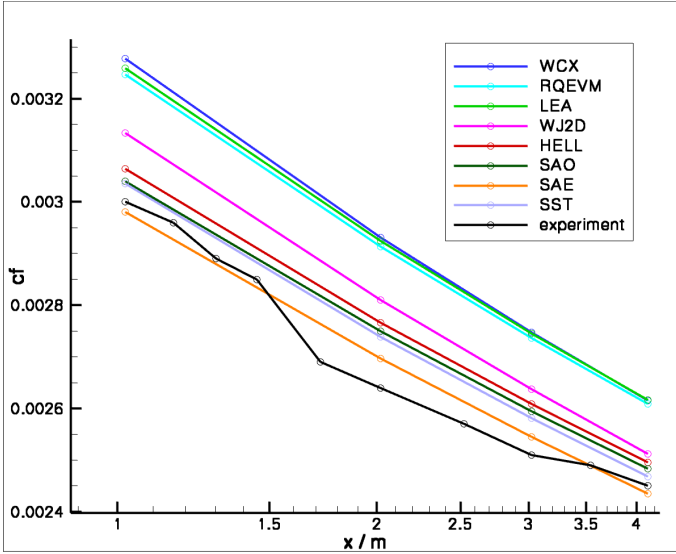


Fig. 9 Flat plate — friction coefficient c_f with variation of the turbulence model in comparison to the experiment

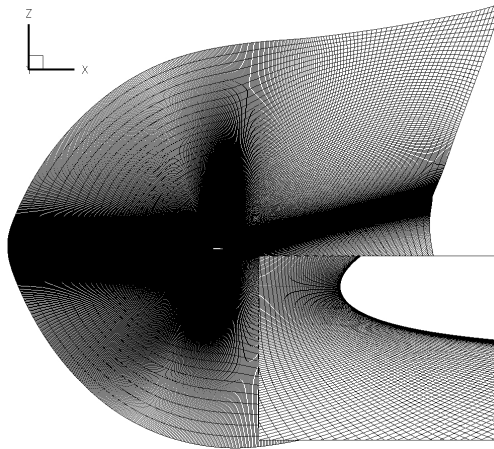


Fig. 10 Grid of the ONERA-A test-case with 530,000 grid points

+/- 3.56% in the lift, +/- 3.43% in the drag coefficient and +/- 7.69% in the position of the separation around the average of the numerical results.

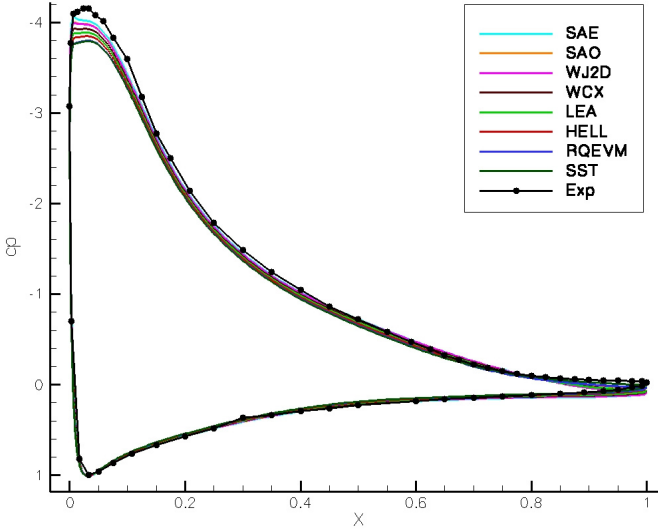


Fig. 11 ONERA-A — pressure coefficient c_p in comparison with the eight different turbulence models and the experiment

Table 1 ONERA-A - lift and drag coefficient, position of pressure induced separation compared to experimental results

	c_L	diff. in %	c_D	diff. in %	separation x/c	diff. in %
Experiment	1.5620		0.0208		0.8945	
HELL	1.4480	-7.30	0.0356	70.96	0.8906	-0.44
LEA	1.4659	-6.15	0.0347	66.92	0.9228	3.16
RQEVM	1.4234	-8.87	0.0351	68.90	0.8883	-0.70
SAE	1.5275	-2.21	0.0337	62.12	0.8312	-7.08
SAO	1.5119	-3.21	0.0350	68.38	0.8078	-9.69
SST	1.4225	-8.93	0.0361	73.62	0.8268	-7.57
WCX	1.4857	-4.89	0.0352	69.08	0.8976	0.35
WJ2D	1.5081	-3.45	0.0342	64.59	0.9423	5.35

Implementation of the Separation Sensor

As already mentioned the development of the boundary layer with different turbulence models leads to different gradients in the friction coefficients. This implies, depending on the position and choice of model, a premature or delayed separation. A constant dependence between model and separation is not identifiable. The user only gets a feedback about an existing separation effect and the critical areas are marked as a surface value in the output files. To detect the appearance of separation

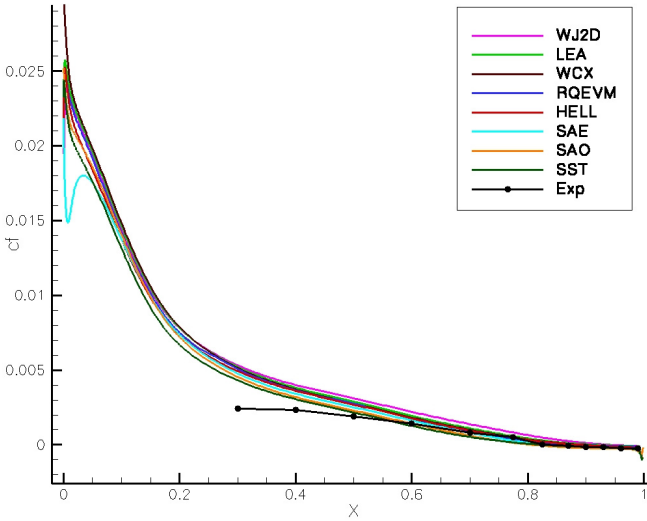


Fig. 12 ONERA–A — friction coefficient c_f in comparison with the eight different turbulence models and the experiment

the sensor scans the surface for characteristic points (first order) of separation or reattachment. For this every surface triangle is tested. Every triangle element which contains two zero-crossings along the edges in the wall-tangential velocities includes a critical point. In the code the velocity at the wall is represented by the components of c_f . These marked elements include hyperbolic points (the beginning or end of a separation line) or rotation points (a vortex separation). To exclude stagnation areas a minimal pressure coefficient of $c_p < 0.5$ is assumed.

3.2.3 Shocks

The correct prediction of flows with high pressure gradients, especially at shocks, also varies with the choice of the turbulence model, both in position and intensity. This again leads to changing values of lift and drag. For the investigation of this influence the 2D test-case RAE 2822 case 9 ($Re = 6.5e6, Ma = 0.73$) with $\alpha = 2.8$ angle of attack and a shock at 55% chord length was applied. The grid is fully structured and consists of 200,000 grid points (fig. 13). It has an orthogonal wall resolution and a highly resolved boundary layer to exclude influences of errors. To achieve a more universal forecast for the computation of the shock position variations of the angle of attack ($\alpha = [1.0deg, 3.2deg]$), the Mach number ($Ma = [0.72...0.76]$) and Reynolds number ($Re = [6.0 \cdot 10^6...10^7]$) were investigated.

The variation of the turbulence models leads to visible variations in the shock position and the level of the c_p and c_f coefficients (fig. 14 & 15). In comparison to the experiment the lift coefficient differs up to 4.6% and the drag coefficient up to 8.9% (e.g. Hellsten). The range between the models is $+/- 3.33\%$ in the lift,

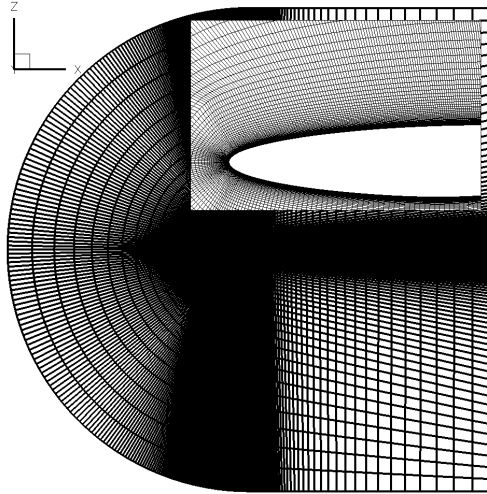


Fig. 13 Grid of the RAE 2822 test-case with 200,000 grid points

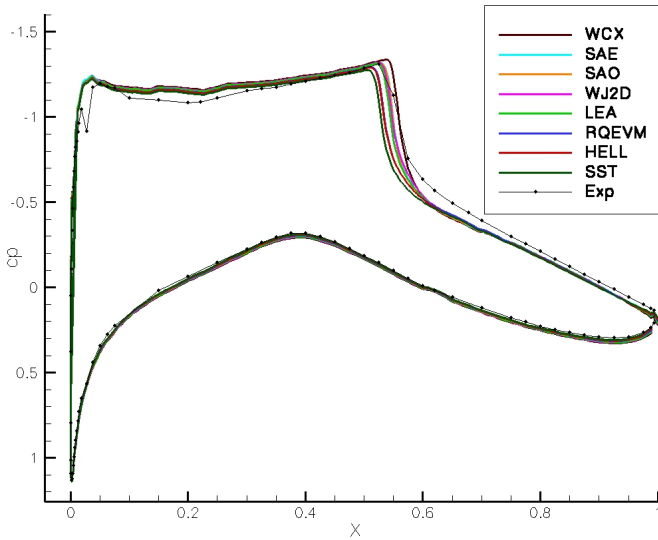


Fig. 14 RAE 2822 case9 — pressure coefficient c_p in comparison with the eight different turbulence models and the experiment

$\pm 7.55\%$ in the drag coefficient and $\pm 2.86\%$ in the shock position around the average of the results. There are obvious dependencies on the choice of the turbulence model and the shock position correlates very well with the lift coefficient. The results have shown that in the most investigated test-cases the prediction of

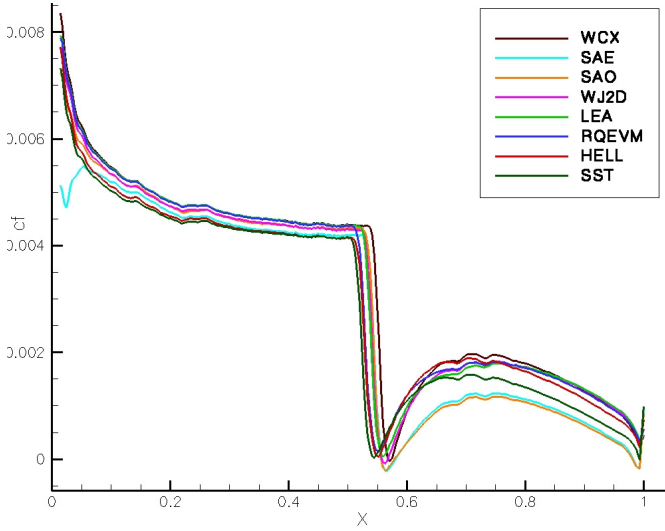


Fig. 15 RAE 2822 case9 — friction coefficient c_f in comparison with the eight different turbulence models

Table 2 RAE2822 case9 - lift and drag coefficient, shock position compared to experimental results

	c_L	diff. in %	c_D	diff. in %	shock x/c	diff. in %
Experiment	0.8030		1.6800E-02		0.5500	
HELL	0.7998	-0.40	1.6410E-02	-2.32	0.5500	0.00
LEA	0.8224	2.42	1.7720E-02	5.48	0.5610	2.00
RQEVM	0.8055	0.31	1.7130E-02	1.96	0.5540	0.73
SAE	0.8244	2.67	1.6240E-02	-3.33	0.5640	2.55
SAO	0.8225	2.43	1.7110E-02	1.85	0.5630	2.36
SST	0.7858	-2.14	1.5730E-02	-6.37	0.5430	-1.27
WCX	0.8400	4.61	1.8300E-02	8.93	0.5750	4.55
WJ2D	0.8284	3.16	1.7770E-02	5.77	0.5650	2.73

the shock position correlates with the turbulence model (left-hand side, fig. 16). In cases where another uncertainty exists, for example a separation bubble, (right-hand side, fig. 16 divergent progress) there is no exact forecast possible for the model dependency. Because of the unchanged order of the models a user feedback that shows only tendency is still possible.

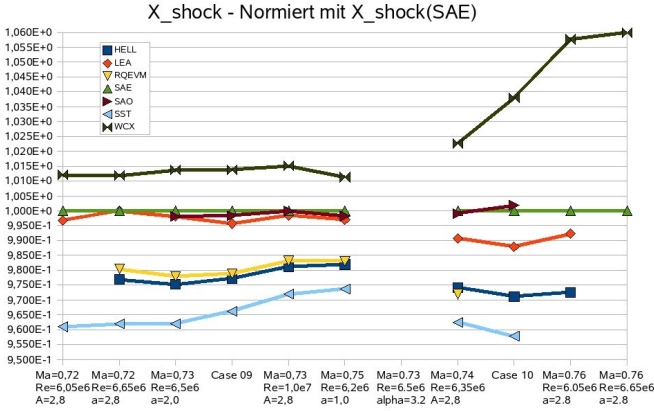


Fig. 16 RAE 2822 case9 — Position of the compression shock normalized by the position of the SAE model with variations of Mach number, Reynolds number and angle of attack

Implementation of the Shock Sensor

To detect a compression shock all grid elements were scanned of characteristic physical values and marked if they exceed a given limit. These values are high pressure gradients in flow direction and a Mach numbers near 1. By projection of the normalized pressure gradients on the velocity a factor for the relative pressure increase was achieved and should come to 30%. The deviation of the Mach number from 1 should be less than 5%. These limits can be adjusted in the TAU solver but represent in all investigated test-cases very good indicators for a shock. The marked areas are available as an output value in the TAU routines.

4 Application of the Sensors

For the usage of the sensors in the standard parameter files some variables were added to activate the best practice guidelines, the visual output and warnings.

4.1 Error Sensors

After activation of the best practice output and declaration of the directory containing the configuration files of the best practice guidelines, the functionality of the individual sensors can be switched on and off. The configuration files are necessary to estimate the errors in the friction coefficients depending on the used model. These files contain a parameterized form of the error function as a polynomial 3rd order

$(p(x) = a_0 + a_1x + a_2x^2 + a_3x^3)$ and the range of validity of this function. The function represents the influence of the error on the friction coefficient and is computed by c_f normalized by the friction in the reference grid $c_{fn} = c_f/c_{f,ref}$. In addition for all errors a warning is given about the resulting influence on the skin friction component of the integral force coefficients. For this the drag coefficient was corrected by the resulting error in the friction and integrated along the wall surface. The computed and the corrected coefficients are compared and the result is written as a feedback in the flow solver output in %.

The error output and prediction are available for the turbulence models listed in Section 4.

Grid Inclination

The graphical surface output of the inclination of the grid is given as a average deviation from the wall normal vector in degree (fig. 17 & 18). Additionally the differences in the friction c_f to the results of the ideal reference grid were written out as an expected error in % ($c_{fn} \cdot 100\%$).

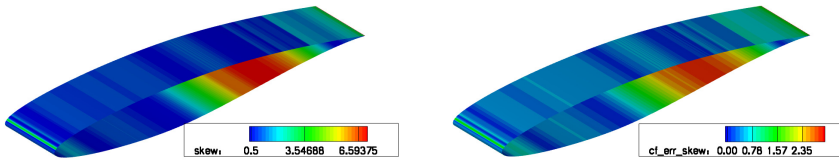


Fig. 17 RAE 2822 case9 — inclination of the grid in the boundary layer in degree (left), error in c_f caused by the inclination in the boundary layer in % (right)

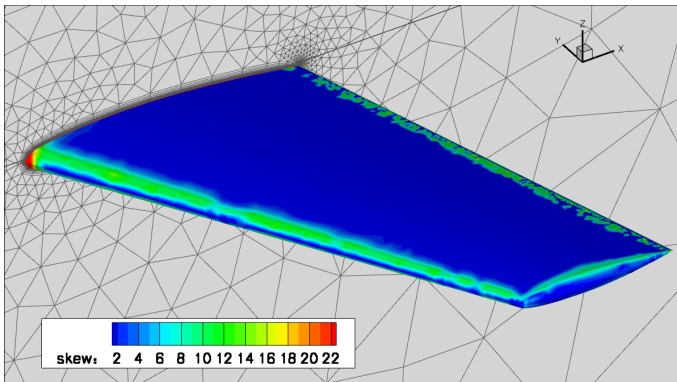


Fig. 18 ONERA M6, very coarse grid — inclination in the boundary layer in degree

Wall Distance y^+

y^+ is a standard surface output value of the flow solver TAU. With that the implemented sensors can calculate the estimated error caused by the wall distance (fig. 19). This represents the predicted difference to the results of a simulation with the reference grid and a very small y^+ in %.

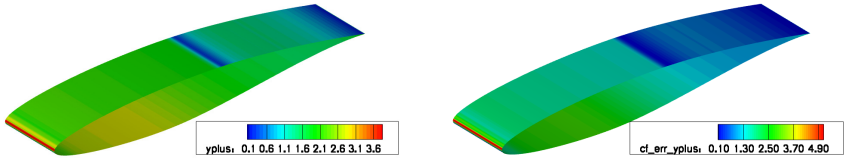


Fig. 19 RAE 2822 case9 — wall distance y^+ (left), error in c_f caused by the wall distance y^+ in % (right)

Grid Expansion Ratio

The graphical output of the maximal grid expansion in the boundary layer is given as a surface value in the TAU routines. Again the differences in the friction c_f to the results of the ideal reference grid with an expansion ratio of 1.1 were written out as an expected error in % (fig. 20 & 21).

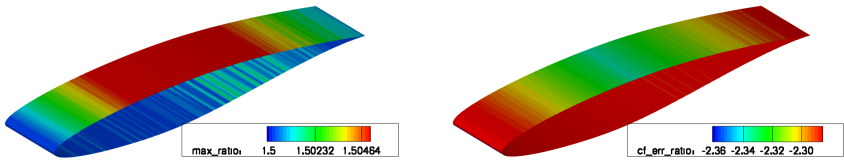


Fig. 20 RAE 2822 case9 — expansion ratio of the grid in the boundary layer (left), error in c_f caused by the grid expansion ratio near the wall in % (right)

4.2 Uncertainty Sensors

For visualization of the uncertainties the areas of a shock based on Mach number and pressure gradients, areas of stagnation and surface elements including 1st order critical points of separation and reattachment are marked in the output routines.

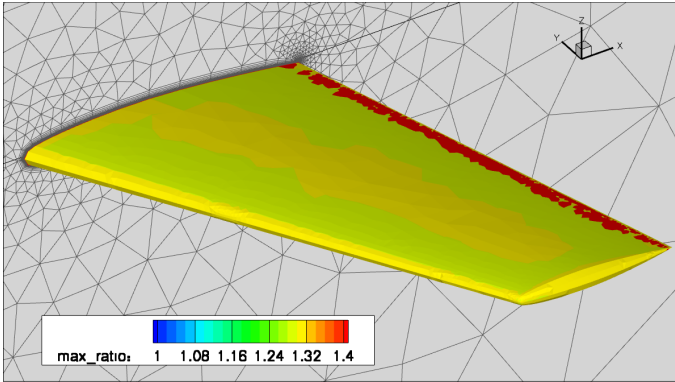


Fig. 21 ONERA M6, very coarse grid — expansion ratio of the grid in the boundary layer

Additionally the number of marked elements, the points in the boundary layers and the boundary layer thickness were written out as further information.

Boundary Layer

As graphical output the number of grid points in the boundary layer and the boundary layer thickness δ in m is given in the flow solver output.

Separation

As graphical output the surface elements including characteristic points of separation and reattachment plus both surface and volume elements with stagnation areas are marked in the flow solver output. Additionally the flow solver can give number of marked elements as a textual feedback.

Shock

As graphical output the volume elements including the characteristics of a shock are marked in the flow solver output. Additionally the flow solver can give a tendential prognosis of the position of the shock depending on the choice of the turbulence model. This is based on the investigations with the 2D RAE test-case.

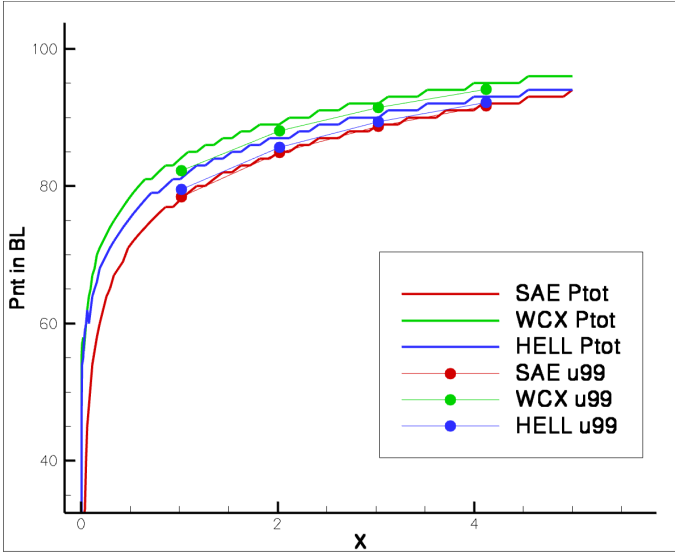


Fig. 22 Flat plate — comparison of the number of points in the boundary layer detected by the sensor and with the boundary layer defined by 99% of the farfield velocity u_{99} with the SAE, Wilcox $k-\omega$ and Hellsten EARSM model

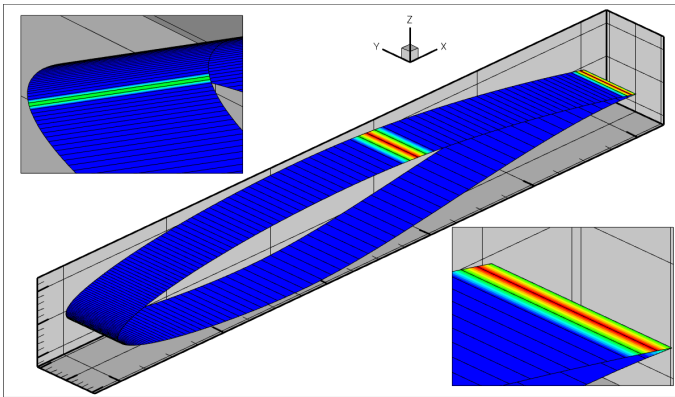


Fig. 23 RAE 2822 case9 — characteristic points of flow separation and reattachment after the shock and near the trailing edge and the stagnation point

5 Applicability on Mixed Errors

To test the applicability of the sensors on mixed errors the test-case RAE 2822 was examined. In figure 26 the friction coefficients of the test-cases with the separated and combined errors are compared with the reference test-case. The errors in the expansion ratio and the wall distance leads to differences in the prediction of the

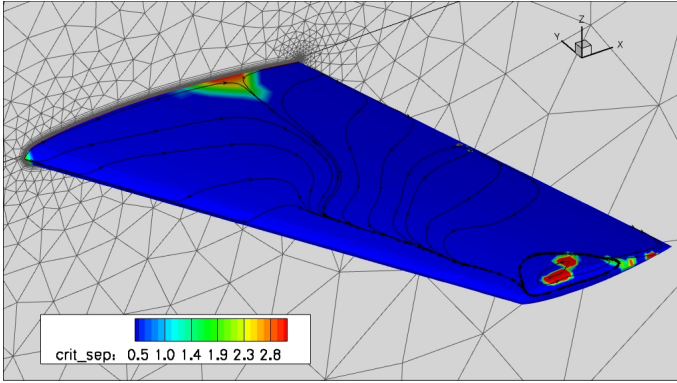


Fig. 24 ONERA M6, coarse grid — characteristic points of flow separation and reattachment and vortex separation at the airfoil tip

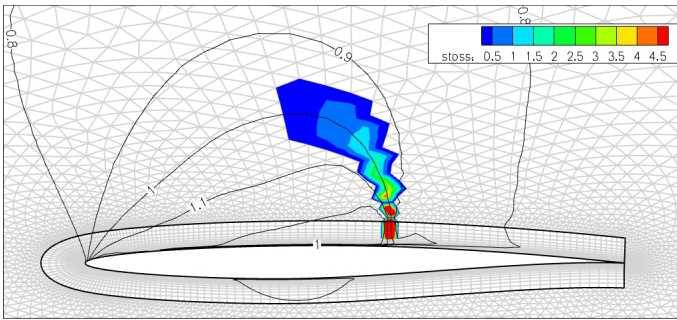


Fig. 25 RAE 2822 case9 — identification of the area of a compression shock

friction coefficient c_f and in case of the mixed error both errors annul each other. With correction of the predicted friction by the error function implemented in TAU the differences are clearly reduced (fig. 27).

6 Industrial Application

To test the sensors on industrial test-cases the generic 3D delta wing VFE-2 and the 3D airfoil SFB-401 were simulated. The delta wing VFE-2 shows a separation of a primary vortex on the leading edge with a reattachement on the wing, which leads to a separation and reattachement of a secondary vortex. On the left-hand side in the figure 28 the critical points of separation and reattachement are shown (red), which presents the separation of the primary vortex on the leading edge and the vortex near the trailing edge very well. The separation of the secondary vortex and the

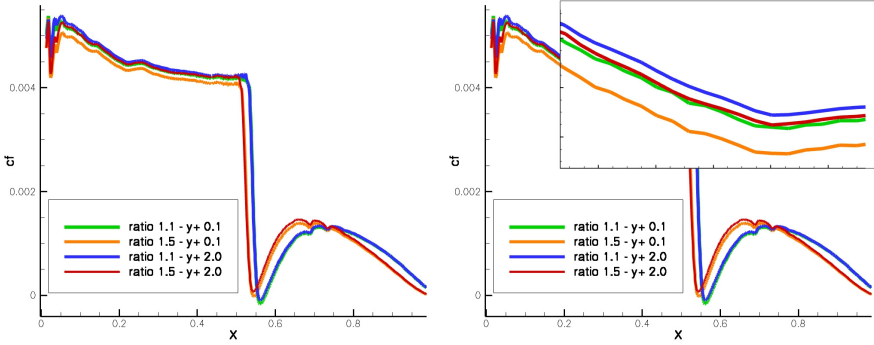


Fig. 26 RAE 2822 case9 — the errors in the expansion ratio and the wall distance leads to differences in the prediction of the friction coefficient c_f , in case of the mixed error both errors annul each other

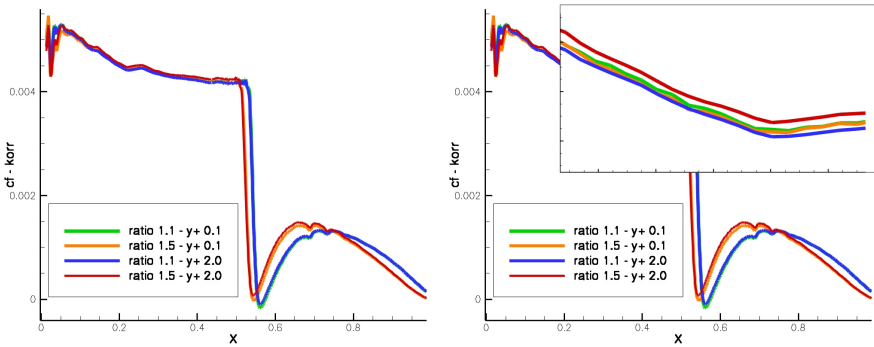


Fig. 27 RAE 2822 case9 — the friction coefficient $c_{f,corr}$ corrected by the error function implemented in TAU, the differences are clearly reduced

reattachment of the primary and secondary vortex are not identified. Both are identifiable in the output of the qualitative variable `sep` for separation structure visualisation in the same figure on the right-hand side. In figure 29 a detailed illustration of the critical points of separation with the output variable `crit_sep` are shown, with the separation of the primary vortex on the leading edge (left) and same small vortices near the trailing edge (right). The sensors for the inclination and the expansion ratio are applied on the test-case SFB-401. The results are shown in figure 30.

7 Field of Application and Its Limits

The analysis and visualization of the boundary layer resolution was developed at the 2D test-case of the flat plate and successfully transferred to $2\frac{1}{2}$ D and 3D airfoil. For this the boundary layer edge was identified at 99.9% of the free stream total pressure. The sensor is only active in this region and interfering effect such as a shock or separation can reduce the boundary layer and so the active region to only a few cells. Because of this the analysis in these regions can vary. Also special test-cases with interacting boundary layers as in high lift configurations are not tested yet. The error mechanisms were investigated in a large range of variations. For these the error estimation was parameterized and is provided for the normalized

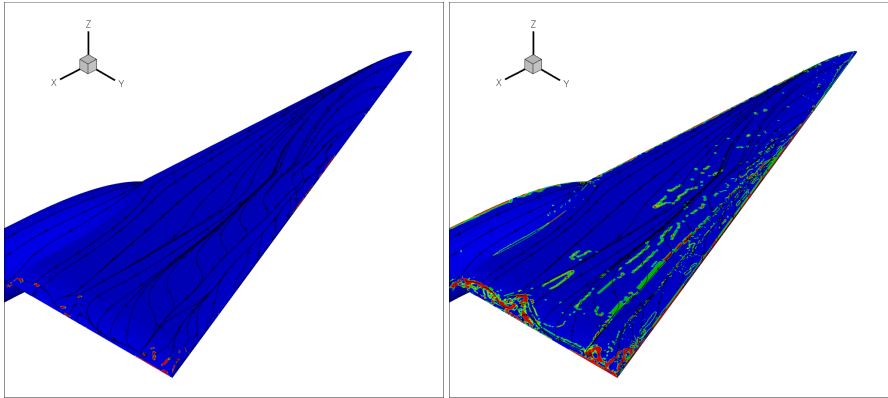


Fig. 28 VFE-2, coarse grid, WJ2D model — critical points of separation and reattachment (red) (left) and qualitative separation structure visualisation (right)

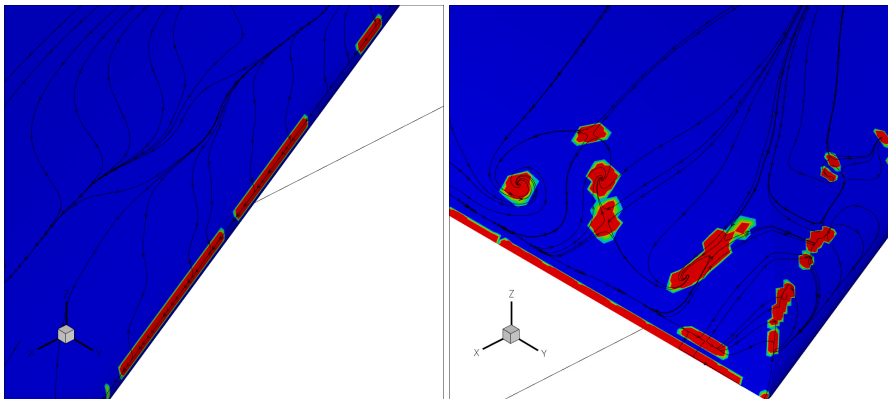


Fig. 29 VFE-2, coarse grid, WJ2D model — a detailed illustration of the critical points of separation with the output variable `crit_sep` (red), the separation of the primary vortex on the leading edge (left) and same small vortices near the trailing edge (right).

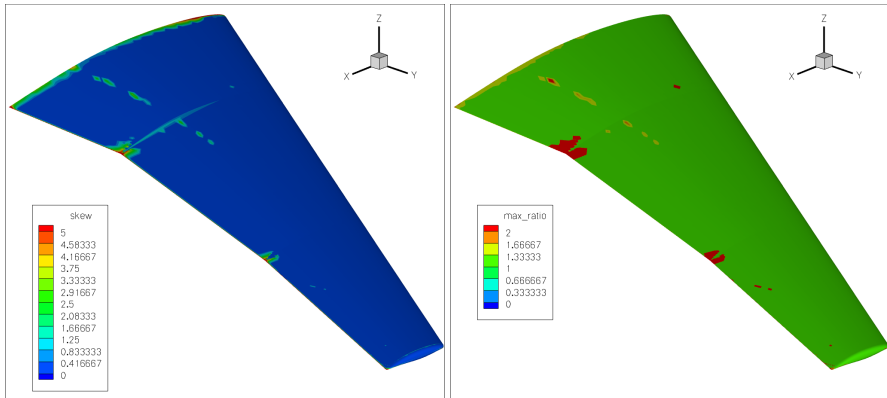


Fig. 30 SFB-401, coarse grid, SAE model — illustration of the average grid inclination in the boundary layer in degree (left) and the maximum expansion ratio in the boundary layer (right)

wall distance in the range of $y^+ = [0.01, 5]$, the grid inclination in the range of $\alpha = [0, +/- 70]$ and the wall normal grid expansion in the range of $r = [1.1, 2.0]$. In the estimations the average inclination angle and the maximum expansion ratio were assumed as constant and only the greatest possible error is given.

References

- [1] Ercoftac Best Practice Guidelines, European Research Community on Flow, Turbulence and Combustion, Special Interest Group on Quality and Trust in Industrial CFD (2000)
- [2] Kozulovic, D.: Modellierung des Grenzschichtumschlags bei Turbomaschinenströmungen unter Berücksichtigung mehrerer Umschlagsarten, Ruhr-Universität Bochum, Fakultät für Maschinenbau, Bochum (2007)

Sensor Controlled Zonal RANS-LES Method

Benedikt Roidl, Matthias Meinke, and Wolfgang Schröder

Abstract. A sensor was developed to detect aerodynamic regions in the computational domain of aerodynamic relevant flows where a common RANS simulation does no longer provide physically reliable results. These regions of the computational domain were then treated with a higher order turbulence model to suppress the modeling error being introduced by standard turbulence models inherent to RANS simulations. The sensor evaluates flow characteristics such as pressure gradients, Reynolds shear stress, and wall shear stress to detect model susceptible regions and to decompose the flow domain in RANS and LES regions. The subsequent approximate integration is performed by a zonal RANS/LES approach which combines the various RANS and LES areas. The sensor and the zonal method are validated by computing the flow over a flat plate, a shock boundary-layer interaction case and a transonic flow over an airfoil.

1 Introduction

To efficiently and accurately determine turbulent flows is still one of the major challenges in computational fluid dynamics. To develop a turbulence model which on the one hand is simple enough to allow an efficient solution and on the other hand, is general and susceptible to describe highly intricate flow phenomena sounds like a contradiction. Numerous classes of turbulence models (algebraic-, one-, two-equation-, RS-models) have been developed until today since the exact description of rotational, fully three-dimensional and time dependent flow is highly complex.

The introduction of the Reynolds averaging process to the Navier-Stokes equations presents the first step of simplification. The complete time-averaged

Benedikt Roidl · Matthias Meinke · Wolfgang Schröder
Institute of Aerodynamics, RWTH Aachen University,
Wüllnerstraße 5a, 52062 Aachen, Germany
e-mail: b.roidl@aia.rwth-aachen.de
<http://www.aia.rwth-aachen.de>

equation of the Reynolds stress tensor consists of different terms describing local and convective parts, production due to the Reynolds stress tensor, dissipation, pressure shear-correlation and diffusion [27]. A closed solution of these transport equations is not possible since they also consist of unknown correlation functions. Thus, semi-empirical equations for turbulent closure are employed. Simplifying the complete contracted time averaged balance equation for boundary layers decreases its complexity since the pressure shear correlation and higher-order terms vanish. This presents a significant simplification of the transport equations. When the second-order normal stress contribution in the production terms and the viscosity linked contributions to the dissipation terms are neglected the final simplified form of the transport equation for the kinetic energy is derived.

The simplifications mentioned above which are applied in one- and two-equation turbulence models will not give a physical answer in complex flows independent of the corresponding discretization level or grid quality. For this reason a higher order turbulence model has to be employed. It can be stated that by applying turbulence models - algebraic, one-equation, two-equation or Reynolds stress models - which include a higher order Boussinesq ansatz - no generally valid trend can be given concerning the prediction quality of specific flow phenomena such as flow separation and reattachment in aerodynamic applications. Moreover, the insufficient quality of the transition prediction constitutes another major uncertainty. A correlation between the quality of turbulence modeling and the variation of flow parameters cannot be provided in a general sense.

These general requirements and limitations in turbulence modeling restrict the applicability of RANS/URANS-methods to simple flow structures. The simulation of attached airfoil flow gives reliable results on skin friction and pressure distributions, however, when high frequency transient flow phenomena occur the quality of the results is questionable. This was the motivation for the development of a sensor which is capable of detecting flow regions of a RANS simulation, where the reliability of the quality of the solution cannot be ensured. In these very regions, a higher order turbulence modeling is necessary leading to the idea of a zonal RANS-LES approach [9].

The development of this sensor requires an exact knowledge of possible physical errors in RANS turbulence models. In this study a sensor was developed to identify the flow regions where RANS simulations produce inaccurate results. These identified flow regions can be treated by a zonal RANS-LES approach.

The article is organized as follows: In section 2 the numerical methods of the flow solver, the sensor and synthetic turbulence generation methods (STGM) are described. Subsequently, in section 3 the results are presented. First, the sensor is validated for three different flows and the susceptibility to critical parts of the sensor is discussed. Then, the various STGM are compared for different configurations in zero-pressure gradient boundary layers. Finally, the fully coupled zonal RANS-LES approach is compared with corresponding full domain LES and RANS solutions for two different cases. First, the case where an oblique shock impinging on a compressible turbulent boundary layer of a flat plate, i. e. the classical shock boundary-layer

interaction problem, is discussed. Second, a transonic flow over an airfoil causing buffet is considered.

2 Numerical Methods

2.1 Large-Eddy Simulation

The three-dimensional unsteady compressible Navier-Stokes equations are solved based on a large-eddy simulation (LES) using the MILES (monotone integrated LES) approach [3]. The vertex-centered finite-volume flow solver is block-structured. A modified AUSM method is used for the Euler terms [18] which are discretized to second-order accuracy by an upwind-biased approximation. For the non-Euler terms a centered approximation of second-order is used. The temporal integration from time level n to $n + 1$ is done by a second-order accurate explicit 5-stage Runge-Kutta method, the coefficients of which are optimized for maximum stability. For a detailed description of the flow solver the reader is referred to Meinke et al. [19].

2.2 Sensor Development

To identify adequate sensor parameters for aerodynamic problems, simple internal flows at variable pressure gradient were investigated. Especially the experimental data of Driver and Johnston [5] was used. In the experiment, an axial incompressible flow over a cylinder is subjected to a variable pressure gradient. It is known that there is a consistent correlation between pressure gradient and local maximum of main Reynolds stress tensor component. The quality of the prediction of the skin friction coefficient and $\langle u'v' \rangle_{max}$ highly depends on the RANS-turbulence model that is used for that kind of flow. Upstream of the boundary layer separation point the local quantities $c_f = \tau_w / (0.5\rho u_\infty^2)$, $c_p = p_w / (0.5\rho u_\infty^2)$ and $\langle u'v' \rangle_{max}$ deviate from the experimental findings, because the wall bounded boundary layer is in a non-equilibrium state [5]. Turbulence models such as algebraic, one- and two equation models do not account for such type of boundary layer flow since for instance turbulent production and dissipation are no longer in the same order of magnitude. The thorough investigation of the flow field yielded the main influence parameters of the sensor; the Clauser-parameter $\beta = \delta_1 / \tau_w \frac{dp}{dx}$, the wall shear stress coefficient c_f , and the local maximum of the main component of the Reynolds shear stress $\langle u'v' \rangle_{max}$. Other parameters such as turbulent production and dissipation could not be used separately because they were not applicable for all kinds of turbulence models. The value ϕ of the sensor is computed by applying the following summation:

$$\phi(c_f, \beta, \langle u'v' \rangle) = T_1(c_f) + T_2(\beta) + T_3(\langle u'v' \rangle) \quad (1)$$

with

$$T_1 = \gamma_1 \frac{1}{\beta_{crit}} \beta, \quad T_2 = \gamma_2 \frac{c_f + 2c_{f,crit}}{c_{f,crit}}, \quad T_3 = \gamma_3 \nabla \left(\frac{\langle u'v' \rangle}{u_\infty^2} \right) \quad (2)$$

where T_1, T_2 and T_3 depend on c_f, β and $\langle u'v' \rangle$ and their user defined critical values ($c_{f,crit}$ and β_{crit}). The values of the prefactor γ_i , that contains reference values being extracted at a zero pressure gradient location in the flow domain, determine the priority of each term in Eq. 2. The calibration of the sensor is done via critical values where the RANS solution is no longer expected to yield reliable solutions. The sensor in this form should not be applied to flows with highly three-dimensional boundary layers or strong vortex interactions. An example configuration for three different flow cases is given in Tab. 1.

2.3 Synthetic Turbulence Generation Methods (STGM)

In this study, the identification of flow regions where RANS should be replaced by a higher order turbulence model represents the first step for the zonal approach. The second step is the proper coupling of RANS and LES flow domains. In order to keep overlapping regions of both computational domains as small as possible, effective mechanisms for turbulence generation have to be applied in LES inflow regions. The turbulent intensities coming from the RANS domain are introduced first at the LES inflow plane via synthetic turbulent eddies (Jarrin *et al.* [12], Batten *et al.* [2]) and controlled further downstream by employing control planes according to Spille and Kaltenbach [24]. The synthetic turbulence generation methods of Jarrin *et al.* and Batten *et al.* were implemented and tested for incompressible and compressible flows.

Method of Jarrin *et al.*

The method of Jarrin *et al.* [12], called synthetic eddy method (SEM), is based on the considering of turbulence as a superposition of coherent structures. These structures are generated over the LES inlet plane and are defined by a shape function which describes the spatial and temporal characteristics of the turbulent structure.

The shape function f_σ that has a compact support on $[-\sigma, \sigma]$ where σ is a length scale which satisfies the normalization condition

Table 1 Suggested critical and reference Values for three different flow types

-	Oblique shock at flat plate	Transonic profile flow	Subsonic profile flow
$c_{f,ref}$	0.002	0.004	0.004
β_{ref}	1	1	1
$c_{f,crit}$	0.0001	0.0001	0.0001
β_{crit}	4	4	4

$$\frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} f_{\sigma}^2 dx = 1 \quad (3)$$

where Δ defines the extent of the domain. A one component velocity signal can then be described by the sum of the contribution $u^{(i)}(x)$ of a turbulent spot i to the velocity field. Let N be the number of prescribed synthetic eddy cores at the inlet and ε_i is a random number within the interval from -1 to $+1$ then the one-dimensional velocity fluctuation component reads

$$u'_j(x, t) = \frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i f_{\sigma}(x - x_i). \quad (4)$$

The generalization of the one-dimensional procedure to time dependent two-dimensional fluctuations is straight forward.

Turbulent length and time scales are determined by the Reynolds shear stress component $\langle u'v' \rangle$ and the turbulent viscosity ν_t ; both are extracted from corresponding RANS simulations. In this work, the Spalart-Allmaras turbulence model [23] was used for the incoming RANS solution. The turbulent time scale can be written as $t = k/\varepsilon$ and the turbulent length scale as $L = t V_b$ with $V_b = \sqrt{k}$ where k and ε stand for the turbulent kinetic energy and turbulent dissipation, respectively. By applying the experimental correlation of Bradshaw *et al.* the turbulent kinetic energy is related to the Reynolds shear stress $\langle u'v' \rangle$ and the turbulent viscosity ν_t which is available from the RANS solution

$$|\langle u'v' \rangle| = \nu_t \left| \frac{\partial u}{\partial y} \right| = a_1 k \quad (5)$$

with $a_1 = \sqrt{c_{\mu}}$ and $c_{\mu} = 0.09$. The turbulent dissipation ε is approximated by the definition of the eddy viscosity from the $k - \varepsilon$ turbulence model (Menter [20])

$$\varepsilon = c_{\mu} \frac{k^2}{\nu_t}. \quad (6)$$

The final flow field at the inlet is constructed from the resulting vortex field of Eq. 4

$$u_i = \bar{u}_i + a_{ij} u'_j \quad (7)$$

where a_{ij} is computed from the prescribed Reynolds stress tensor applying a Cholesky decomposition.

Method of Batten et al.

The second method was introduced by Batten *et al.* [2] based on the work of Smirnov [21] and initially developed by Kraichnan [16]. To create a three-dimensional, unsteady velocity field at the inflow plane of the LES region, velocity components are constructed using a sum of sines and cosines with random phases and amplitudes. The intermediate velocity, v_i , reads

$$v_i(x_j, t) = \sqrt{\frac{2}{N}} \sum_{n=1}^N [p_i^n \cos \hat{d}_j^n \hat{x}_j^n + \omega^n \hat{t} + q_i^n \sin \hat{d}_j^n \hat{x}_j^n + \omega^n \hat{t}], \quad (8)$$

where \hat{x} are spatial coordinates being normalized by turbulent length- and time scales. These scales are reconstructed from the incoming RANS solution via Eqs. 5 and 7. The amplitudes of the signal are calculated by

$$p_i^n = \varepsilon_{ijk} \zeta_j^n d_k^n, \quad q_i^n = \varepsilon_{ijk} \xi_j^n d_k^n \quad (9)$$

where ξ and ζ are equal to $N(0, 1)$ and $\hat{d}_j^n = d_j^n V / c^n$. The wave number $d_i^n = N(0, 0.5)$ is elongated by the following relation according to Batten [2]:

$$c^n = \sqrt{\frac{3}{2} \frac{u'_l u'_m}{u'_k} \frac{d_l^n d_m^n}{d_k^n d_k^n}}. \quad (10)$$

In Eq. 8 the random frequencies ω^n are taken from the normal distribution $N(1, 1)$. Like in the method of Jarrin *et al.* the synthetic turbulent fluctuation field is finally reconstructed using the Cholesky decomposition. The methods are suitable in incompressible flows. In compressible flows, however, the velocity fluctuations are coupled with the density field. Thus, Morkovin's hypothesis is applied to relate density and velocity fluctuations by assuming that the pressure fluctuations over the inflow plane are negligible

$$\frac{\rho'}{\bar{\rho}} = (\gamma - 1) M^2 \frac{u'}{\bar{u}}. \quad (11)$$

Spille-Kaltenbach Control Planes

Synthetic turbulent methods provide a reasonable first estimate of the fluctuating turbulent velocity field at the LES inlet. Downstream of the inlet, however, many of the relevant turbulent scales may have been dissipated retarding the transition to fully turbulent flow. Local control planes which introduce a volumetric forcing term to the Navier-Stokes equations regulate the turbulent production in the shear stress budget [24]. As discussed, for example, in the work of Keating *et al.* [14] or Zhang *et al.* [29], local flow events such as bursts and sweeps are enhanced or damped by the local forcing thus contributing to the Reynolds shear stress $\langle u'v' \rangle$

$$e(y, t) = \langle u'v' \rangle^*(x_0, y) - \langle u'v' \rangle^{z,t}(x_0, y, t) \quad (12)$$

where $\langle u'v' \rangle^*$ is the target Reynolds shear stress at the control plane which is provided by the RANS solution and $\langle u'v' \rangle^{z,t}$ is the current Reynolds shear stress in the LES domain which is averaged over the spanwise direction and time. For the time average a window function with a time constant equal to $\approx 100\delta_0/u_\delta$ is used. The force magnitude is given by

$$f(x_0, y, z, t) = r(y, t) [u(x_0, y, z, t) - \langle u \rangle^{z,t}] \quad (13)$$

with

$$r(y,t) = \alpha e(y,t) + \beta \int_0^t e(y,t') dt' . \quad (14)$$

The proportional part is the main contributor to the force when the error e in Eq. 12 is high at the beginning of the simulation. Proceeding in time, the integral part gives the force the necessary response to enhance or damp the local flow events. The constants α and β were set to 10 and 25 respectively, to ensure on the one hand, a rapidly decreasing error e and on the other hand, a stable simulation process. In subsequent sections the STGM of Jarrin *et al.* combined with the control plane approach is referred to as 'zonal SEM' and the STGM of Batten *et al.* combined with the control plane approach is referred to as 'zonal Batten'.

3 Results

3.1 Sensor Validation

Three different test cases for sensor calibration are considered in detail in the following sections showing the influence of the user defined critical values on the sensor value ϕ . The configuration of these simulations is given in Tab. 2.

In Fig. 1 the application of the sensor on a flow case, where a shock wave impinges on a compressible turbulent boundary layer (SWBLI) causing a local flow separation, is presented. The Spalart-Allmaras turbulence model [23] (referred to as 'S-A') was applied for this simulation. Regardless the computational configuration, the sensor value ϕ exceeds 1 where the flow is separated. Downstream of the separation the boundary layer returns to an equilibrium state. Depending on the user defined critical Clauser parameter the suggested corresponding LES domain is smaller (for $\beta_{crit}=8$) or larger (for $\beta_{crit} = 2$). In section 3, this particular case is regarded thoroughly using RANS, LES and zonal RANS-LES computations.

Fig. 2 presents the resulting sensor values ϕ for the upper side of the DRA23032-profile for two different time steps. The Baldwin-Lomax turbulence model [11] (referred to as 'B-L') was applied for this case. The sensor value at time step t_0 exceeds the allowed limit of 1 at $0.55 c$ where the shock is located. However, the flow does not separate before passing $0.65 c$ (not shown here). The high pressure gradient

Table 2 Test cases for Sensor validation

-	Transonic airfoil flow	Oblique shock on flat plate	Subsonic airfoil flow
Re	$2.6 \cdot 10^6$ (reference c)	19000 (reference δ_0)	$2.1 \cdot 10^6$ (reference c)
Ma	1	1	1
α	3°	-	13°
T-M	B-L	S-A	S-A

and the rapid decline of c_f lead to an increase of ϕ . Downstream of the shock position the sensor value ϕ does not fall below 1 for more than several percent chord due to the reason that the boundary layer is locally separated and/or in a non-equilibrium state. The skin friction coefficient and Clauser parameter are the major contributors to the sensor value ϕ . At t_0 the shock position is located further upstream which is in fact the most upstream location of the shock. This shows that the size of the intended LES domain should be evaluated over a time span of several shock oscillations. This case is also thoroughly investigated with in section 3.

Figure 3 shows the results of the sensor application on the case of a subsonic profile at high angles of attack 4 (see Tab. 2 for two different values of $c_{f,crit}$). The turbulence model of Spalart and Allmaras was used for this case. Due to a laminar separation bubble at about $0.12c$ the sensor value ϕ is higher than 1 up to $0.20c$. For a user defined critical $c_{f,crit}$ of $1 \cdot 10^{-4}$ the sensor predicts the end of the confidence domain at about $0.75c$ which is located upstream of the experimentally determined separation point of $0.85c$. For $c_{f,crit} = 5 \cdot 10^{-4}$ the sensor value ϕ exceeds the value 1 at about $0.60c$, thus the proposed LES domain is larger compared to the case using the lower $c_{f,crit}$ value. This is due to the more or less equal contribution of c_f , β and Reynolds shear stress gradient $\nabla \langle u'v' \rangle$ to the sensor value ϕ (see Eq. 1). This case was also extensively investigated by Celić and Hirschel 4 where the results agree with the present RANS simulation. The presented RANS simulation indicates that all the investigated turbulence models (algebraic, one, two-equation models and RSM) show an unsatisfying behavior from $0.7c$ to the trailing edge due to the steep decline of c_f and the beginning increase of $\langle u'v' \rangle$.

These three examples demonstrate that the sensor provides a comprehensible interpretation about the confidence domain of RANS solutions for different aerodynamic flow configurations, i.e., various free stream Mach numbers and Reynolds numbers. Most shock boundary-layer interaction problems are transient problems with a time dependent behavior of i.a. c_f , β , and $\langle u'v' \rangle$. As shown in the transonic flow case, the extreme positions of the critical sensor values ϕ at the upper side of the airfoil are used to span the flow field where the confidence in the RANS solution is low.

3.2 Validation of STGM

A zero-pressure gradient boundary layer was investigated and compared to reference solutions using two synthetic turbulence methods, based on the controlled forcing approach downstream of the inlet. Four simulations were carried out: a full domain LES (referred to as 'full LES'), a full domain RANS, and two synthetic turbulence-LES simulations with controlled forcing. All four cases were computed with the same flow and numerical configuration, $M_\infty = 0.4$, $Re_\delta = 10000$. The numerical details are given in Tab. 3.

The reference full LES was computed using the rescaling method according to El-Askary *et al.* 8. The full domain RANS calculation based on the Spalart-Allmaras model (referred to as 'RANS S-A') was performed for comparison

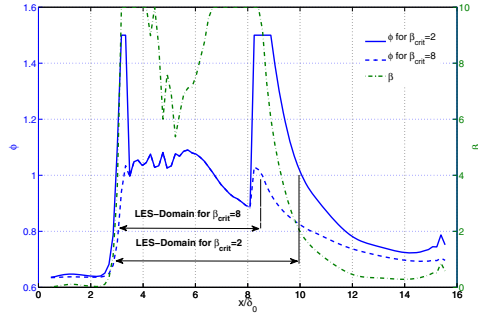


Fig. 1 Sensor value ϕ for two different critical Clauser parameters β_{crit} for the case of the oblique shock at flat a plate

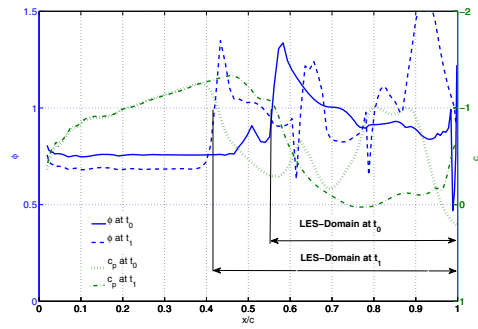


Fig. 2 Sensor value ϕ for two time steps t_0 and t_1 for transonic profile flow

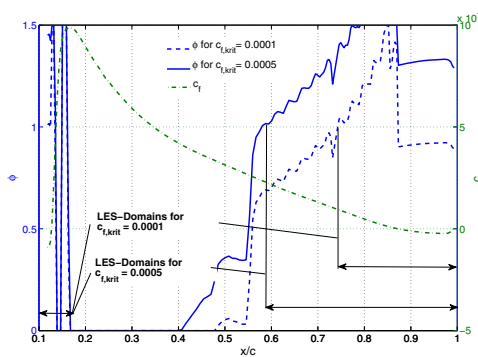


Fig. 3 Sensor value ϕ for two different critical $c_{f,crit}$ for subsonic profile flow

Table 3 Numerical details for the simulation of turbulent boundary layer at $Ma = 0.4$ and $Re_\delta = 10000$ for LES and RANS solutions

	streamwise x	wall normal y	spanwise z
Domain size in δ_0	15	5	0.7
Grid points	484	65	49
Resolution, wall units	$\Delta x^+ \approx 20$	$\Delta y_{min}^+ \approx 0.8$	$\Delta z^+ \approx 12$

purposes and to provide the target data for the synthetic turbulence inlet and for the control planes. In the zonal simulations the control planes are distributed over a length of one boundary layer thickness δ_0 .

Figure 4 (left) compares the evolution of the wall friction coefficient c_f for all four flow cases. The solutions of full LES and full RANS do not differ much regarding the wall shear stress for this simple zero-pressure gradient boundary layer. Thus, the applied rescaling method at this numerical configuration is valid for LES and RANS simulations and poses no difficulty to match them at the beginning of the computational domain. Both, the zonal SEM and the zonal Batten approach, show, despite their fundamental differences in their formulations, a comparable required length until they converge to the full LES solution. The van Driest velocity profiles obtained at $x/\delta_0 = 5$ which are presented in Fig. 4 (right) show that both STGM produce the expected asymptotic near-wall behavior of a turbulent flow, but the results differ somewhat at the edge of the boundary layer.

Figure 5 depicts the turbulent kinetic energy k and Reynolds shear stress component $\langle u'v' \rangle$ at two different locations downstream of the inlet. It is shown that the flow generated by the zonal Batten approach undergoes a slight laminarization process downstream of the interface but the control planes increase the turbulent shear stress budget to the full LES level. The turbulent structures generated by the zonal SEM approach do not dissipate downstream of the inlet but the control planes introduce a local overshoot of the turbulence level which decreases to the full LES turbulence level at around $5 x/\delta_0$. By improving the response of the control planes to the local flow events, the turbulent shear stress level of the full LES could be reached within one boundary layer thickness δ_0 .

λ_2 -contours (Jeong *et al.* [13]) of both zonal cases and the full LES are visualized in Fig. 6. It is shown that the structures which are introduced into the domain by the zonal SEM approach are not dissipating. The structures at the inlet of the case computed with the zonal Batten approach fade away and the control planes downstream of the inlet have to enhance the locally rare events like turbulent bursts and sweeps to reach the turbulence level of the full LES computation.

Due to the low Reynolds number for this case it was expected that the 'artificial' turbulence would dissipate at the beginning of the domain to develop 'physical' turbulence further downstream after the transition process at about $x/\delta_0 \approx 10$. However, Figs. 4 (left) and 5 (left) show that when the zonal SEM approach is used the Reynolds shear stress $\langle u'v' \rangle$ does not decrease below the level of the full domain

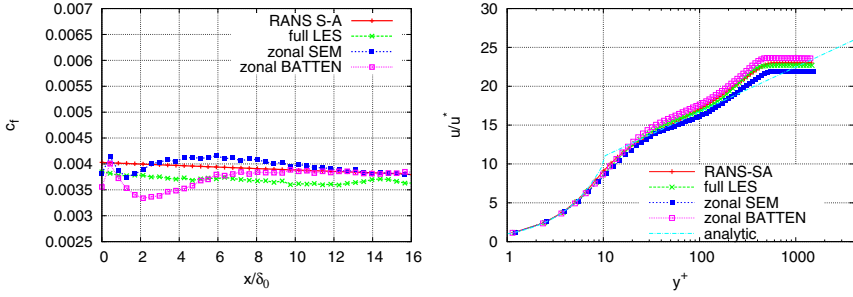


Fig. 4 Evolution of skin friction coefficient c_f (left) and van Driest velocity profile (right) for boundary layer flow at $Ma = 0.4$ and $Re_{\delta_0} = 10000$ for different computational configurations

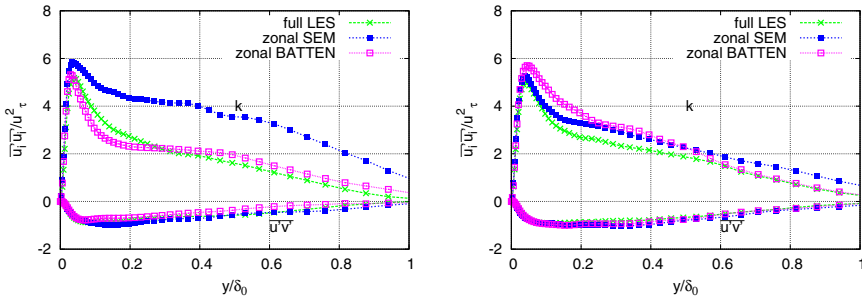


Fig. 5 Turbulent kinetic energy k and Reynolds shear stress $\langle u'v' \rangle$ at $x/\delta_0 = 1$ (left) and at $x/\delta_0 = 5$ (right)

LES but generates an overshoot of turbulent kinetic energy. The zonal Batten approach tends to provide a lower Reynolds shear stress level which is to be increased by the control planes which are located downstream of the inlet. For the following computations the zonal SEM approach is used since the quality of the results is acceptable and it is computationally less expensive than the zonal Batten approach.

A compressible zero-pressure gradient boundary layer ($Ma = 2.4$, $Re_{\delta_0} = 52000$) was also investigated to evaluate the efficiency of STGM in compressible flows. A full LES simulation was used as reference and a full RANS simulation provided targets for the zonal RANS-LES solution. In this case just the zonal SEM approach was used and evaluated concerning two different control plane configurations. The first configuration uses one single control plane (referred to as '1 c-p') which is located at $x/\delta_0 = 0.7$. The second configuration applies four control planes (referred to as '4 c-p') which are located between $x/\delta_0 = 0.3$ and $x/\delta_0 = 2$. Numerical details are given in Tab. 4.

The downstream evolution of the skin friction coefficient c_f is shown in Fig. 7 (left). It seems that the zonal SEM solution with one control plane already converges at about $x/\delta_0 = 1.5$ whereas the case with four control planes still regulates

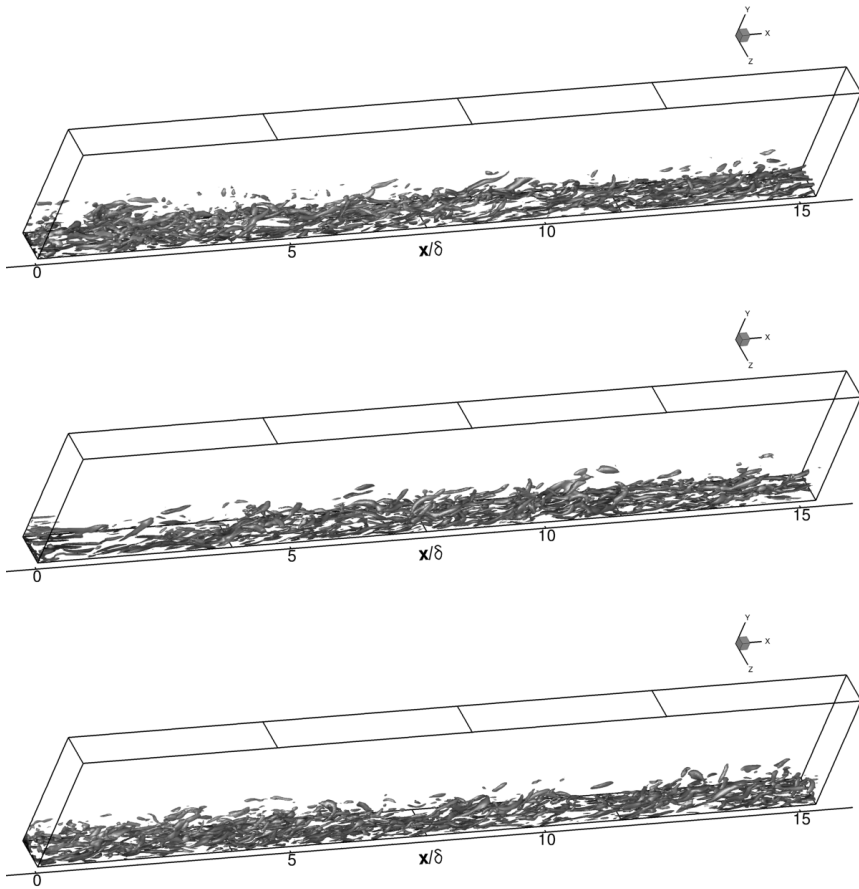


Fig. 6 λ_2 structures of a mildly compressible flat plate boundary layer flow computed by (top) a full LES, (middle) using zonal Batten approach and (bottom) applying the zonal SEM ansatz

Table 4 Numerical details for the computation of a turbulent boundary layer at $Ma = 2.4$ and $Re_\delta = 52000$ for LES and RANS solutions

	streamwise x	wall normal y	spanwise z
Domain size in δ_0	10	3	0.7
Grid points	140	65	49
Resolution, wall units	$\Delta x^+ \approx 20$	$\Delta y_{min}^+ \approx 0.8$	$\Delta z^+ \approx 12$

the shear stress budget. However, Fig. 8 compares the turbulent kinetic energy k and Reynolds shear stress $\langle u'v' \rangle$ at position $x/\delta_0 = 1$. A second spurious peak is exhibited at $y/\delta_0 = 0.7$ for the zonal SEM simulation with one control plane. This

distribution of turbulence energy is due to the presence of a low frequency mode which is introduced at the inlet and may survive at least the first control plane. Toubert *et al.* [25] observed the same phenomenon when applying synthetic turbulence techniques to compressible boundary layers. This might occur due to the high Reynolds and Mach numbers used here which stabilizes the outer mode. However, this mode was not found in subsonic boundary layers. When four control planes are used the flow already passes two control planes and the spurious peak in the wake region of the boundary layer at position $x/\delta_0 = 1$ is somewhat more damped. At position $x/\delta_0 = 5$ both control plane configurations show no trace of this second peak in the turbulent kinetic energy k . From these results it is suggested to apply more than one control plane in supersonic boundary layers at high Reynolds numbers to avoid the large wave length mode introduced by the STGM and thus to keep the transition region as small as possible. The van Driest velocity profiles obtained at $x/\delta_0 = 5$ in Fig. 7 (right) show that both STGM produce the expected asymptotic near-wall behavior of a turbulent flow, but the results differ somewhat at the edge of the boundary layer.

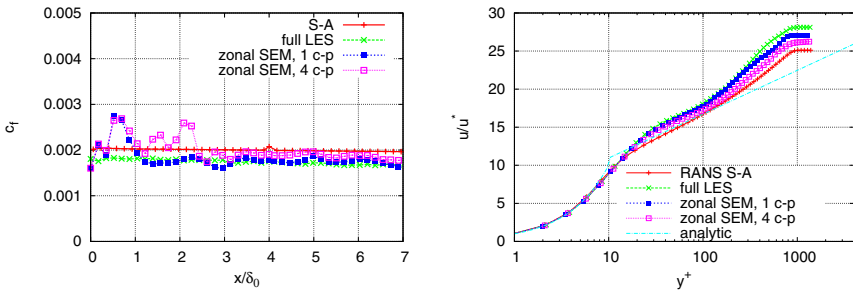


Fig. 7 Evolution of skin friction coefficient c_f (left) and van Driest velocity profile (right) for boundary layer flow at $Ma = 2.4$ and $Re_{\delta_0} = 52000$ for different computational- and control plane configurations

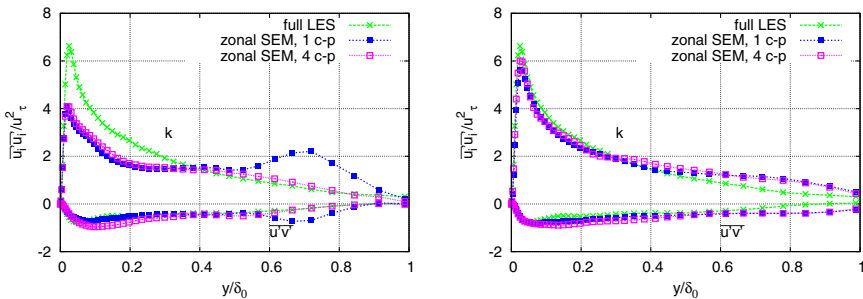


Fig. 8 Turbulent kinetic energy k and Reynolds shear stress $\langle u'v' \rangle$ at $x/\delta_0 = 1$ (left) and at $x/\delta_0 = 5$ (right)

3.3 Oblique Shock on Flat Plate (SWBLI)

In section 3.1 the sensor was tested for an oblique shock impinging the turbulent boundary layer of a flat plate. This case was investigated experimentally by Dussauge *et al.* [6] [7] with the corresponding configuration of $Re_{\delta_1} = 19300$, $Ma = 2.3$, and an oblique shock angle $\alpha = 8^\circ$. In the following three simulations are analyzed: A full domain RANS simulation with Spalart-Allmaras turbulence modeling was computed for the sensor application to determine the size of the computational domains for the zonal RANS-LES simulation. For comparison purposes a full LES was simulated and finally, a fully coupled zonal RANS-LES approach to validate the zonal concept. The LES domain indicated by the sensor evidenced in Fig. 1 is slightly extended since the synthetic turbulence generation needs additional space in the streamwise direction to evolve properly. Figure 9 presents the computational setup of the fully coupled RANS-LES simulation emphasizing the computational domains of RANS and LES as well as four control planes downstream of the LES inlet. The computational set up of the full LES simulation and of the LES part of the zonal RANS-LES simulation are given in Tab. 5 and Tab. 6 respectively. The number of grid points for the zonal RANS-LES simulation is more than 50 % lower compared to the number of points of the full domain LES simulation.

Periodic boundary conditions are used in the spanwise direction and a no-slip, adiabatic condition is set at the wall. The top boundary uses characteristic boundary conditions to minimize spurious reflections from the other boundaries. The oblique shock is introduced at the inlet boundary applying the Rankine-Hugoniot conditions. The inflow condition of the RANS part uses the rescaling method of El-Askary *et al.* [8] while the inflow of the LES was set to the above described zonal SEM approach. At the LES outflow the condition of König *et al.* [15] was applied to ensure a proper transition of the turbulent flow properties from the LES to the RANS domain.

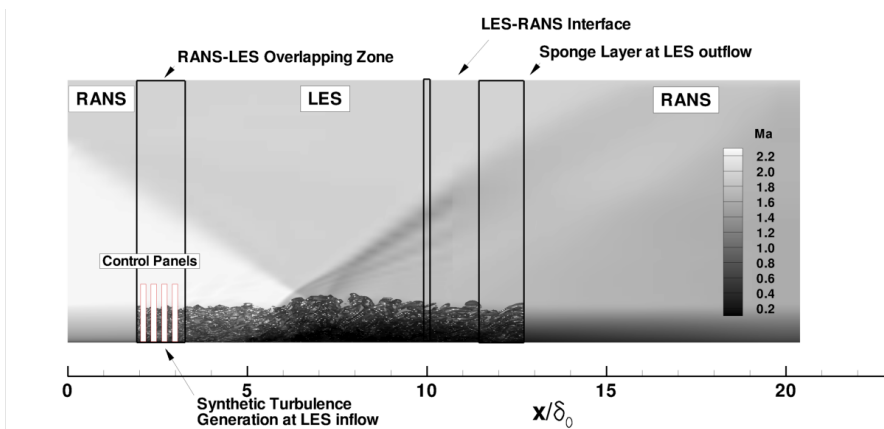


Fig. 9 Computational setup of fully coupled RANS-LES simulation of SWBLI flow case

Table 5 Numerical details for the full LES simulation for the SWBLI case

	streamwise x	wall normal y	spanwise z
Domain size in δ_0	20	6.3	0.6
Grid points ($3.9 \cdot 10^6$)	512	153	50
Resolution, wall units	$\Delta x^+ \approx 35$	$\Delta y_{min}^+ \approx 1.1$	$\Delta z^+ \approx 18$

Table 6 Numerical details for the LES domain of the zonal RANS-LES simulation of the SWBLI case

	streamwise x	wall normal y	spanwise z
Domain size in δ_0	20	6.3	0.6
Grid points ($1.9 \cdot 10^6$)	256	153	50
Resolution, wall units	$\Delta x^+ \approx 35$	$\Delta y_{min}^+ \approx 1.1$	$\Delta z^+ \approx 18$

A sponge zone at the LES outflow ensures the damping of turbulent fluctuations. The streamwise evolution of the skin friction coefficient c_f is displayed in Fig. 10 (left). First, it is noticeable that the RANS, i.e., full RANS and zonal RANS part, and the full LES solution differ for this zero-pressure gradient boundary layer. This difference is caused by different grid resolutions of the full LES and the RANS domain. For all cases the onset of flow separation is located at about $x/\delta_0 = 4.5$. The experimental results of Dussauge *et al.* show an increased c_f level at the inlet compared to the full LES and predict the separation point slightly more upstream. The results of the full LES and the zonal RANS-LES regarding the length of the separation bubble are in good agreement with the LES results of Garnier *et al.* [11]. Downstream of the interaction zone the increase of c_f is very similar between experiments, full LES and especially the zonal RANS-LES solution where the LES to RANS transition method of König *et al.* [15] proved to provide good results. One can notice that the result quality of time averaged c_f of LES and zonal LES is clearly superior to that obtained by a RANS approach in the same configuration.

Figure 10 (right) compares the van Driest velocity profiles obtained at $x/\delta_0 = 3.3$ which is located slightly upstream of the interaction zone. All computations show the expected asymptotic near wall behavior. However, the logarithmic region of the full LES computation lies somewhat above the analytic result because the streamwise grid resolution. The corresponding profile of the zonal RANS-LES solution is located in between that of the full RANS and that of the full LES as it just passes the control planes and hence, still contains the shear stress profile of the target RANS values.

In Fig. 11 (left) the comparison of PIV results [6] with the zonal RANS-LES outcome of the wall-normal velocity fluctuations is given. At position $x = 280\text{mm}$ ($x/\delta_0 = 3.2$), which is about two boundary layer thicknesses downstream of the PIV measurement position, the fluctuations show a higher intensity at the middle of the boundary layer than the corresponding PIV measurements. This is due to the

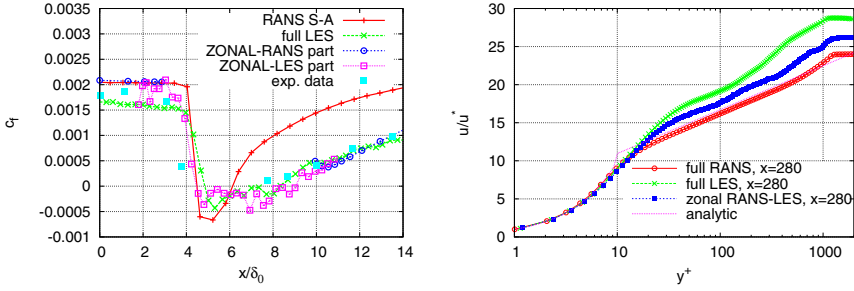


Fig. 10 Longitudinal evolution of skin friction coefficient c_f (left) and van Driest velocity profile (right) for SWBLI flow case for different computational- and control plane configurations

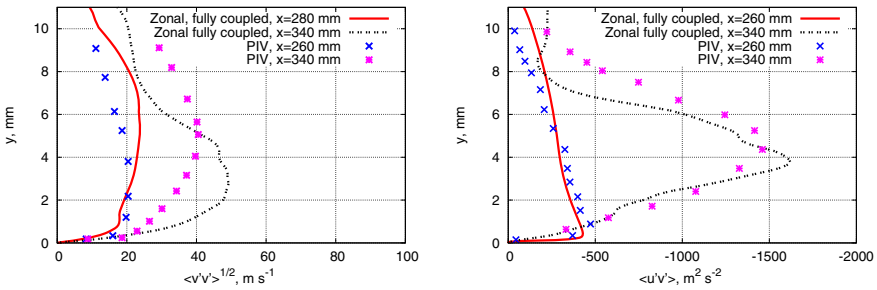


Fig. 11 Comparison between PIV [6] and zonal RANS-LES of the *rms*-wall-normal velocity fluctuations (left) and of the Reynolds shear stress (right)

STGM at the inlet which assigns too much energy in the outer part of the boundary layer. At position $x = 340mm$ ($x/\delta_0 = 9$) the high energy fluctuations which were introduced by the STGM at the inlet in the outer part of the boundary layer did not survive the interaction region and the fluctuation level decreases below that of the PIV measurements.

Figure 11 (right) represents a similar comparison for the Reynolds shear stress. The distribution at $x = 260mm$ ($x/\delta_0 = 1$) shows a good agreement between the zonal RANS-LES results and the PIV measurements. Downstream of the interaction zone, at $x = 340mm$ ($x/\delta_0 = 9$), the results of the zonal RANS-LES agree again fairly well with the PIV measurements. The λ_2 -structures at the interaction zone of the oblique shock and turbulent boundary layer for the full LES and the zonal RANS-LES simulation are depicted in Fig. 12. Large coherent structures are formed at the proximity of the impinging shock which transport the maximum of the turbulent kinetic energy towards the center of the boundary layer. Such turbulent features are evident in the full LES and the zonal RANS-LES solution.

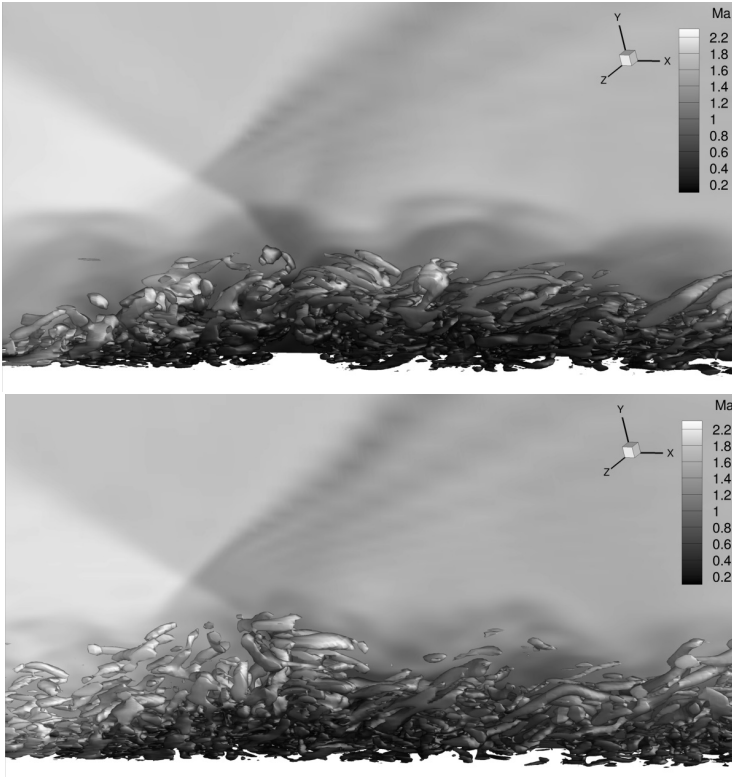


Fig. 12 λ_2 structures in the shock boundary-layer interaction zone computed by (top) a full LES and (bottom) a fully coupled zonal RANS-LES

3.4 DRA2303 Transonic Profile

The DRA2303 transonic airfoil [10] was chosen as the aerodynamic reference case for the buffet phenomenon. Associated with buffet are self-sustained shock wave oscillations on airfoils at transonic flow. The flow configuration, which was to lie well within the buffet boundaries, was chosen with $Ma = 0.72$, $Re = 2.6 \cdot 10^6$ and $\alpha = 3^\circ$. For this flow configuration, experimental data is not yet available.

In this work the buffet is computationally targeted with three different simulations: a full domain RANS-simulation based on the Baldwin-Lomax turbulence model [1], a full domain LES which constitutes the reference solution and a fully coupled zonal RANS-LES solution. First, the full LES solution of the DRA2303 is investigated. The RANS simulation, which was also a calibration case for the sensor in section 3.1, is compared with the reference LES solution. Finally, a fully coupled zonal RANS-LES solution yielding preliminary results is presented. The numerical details of the full LES solution and the zonal RANS-LES are given in Tab. 7 and Tab. 8, respectively. Similar to the SWBLI case the number of required grid points

Table 7 Numerical details for the full LES simulation

	streamwise x	wall normal y	spanwise z
Domain size in c	20	20	0.021
Grid points ($30.4 \cdot 10^6$)	2364	130	99
Resolution, wall units	$\Delta x^+ \approx 100$	$\Delta y_{min}^+ \approx 1.0$	$\Delta z^+ \approx 20$

for the zonal RANS-LES simulation contains less than 50 % of the grid points used in the full domain LES simulation.

The focus of this numerical investigation using a full RANS simulation was on the evaluation of the reduced frequency $\omega^* = \omega c / U_\infty$ for the given flow configuration. In Fig. 13 the instantaneous pressure coefficients c_p and the average pressure coefficient fluctuation at the upper side of the airfoil for the full RANS simulation are shown. At this flow configuration the extension of the horizontal shock movement is about $0.25 c$. High level pressure fluctuations are found downstream of the shock, especially near the trailing edge where the strength of these fluctuations even exceeds the pressure amplitude of the shock.

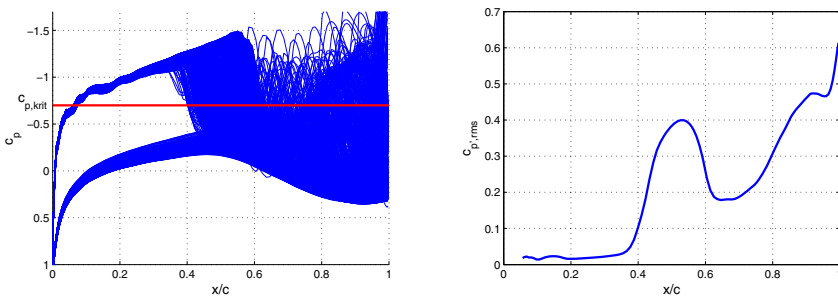


Fig. 13 Fluctuating wall pressure coefficient c_p (left) and corresponding rms -values at the upper side of the profile (right) for a RANS simulation

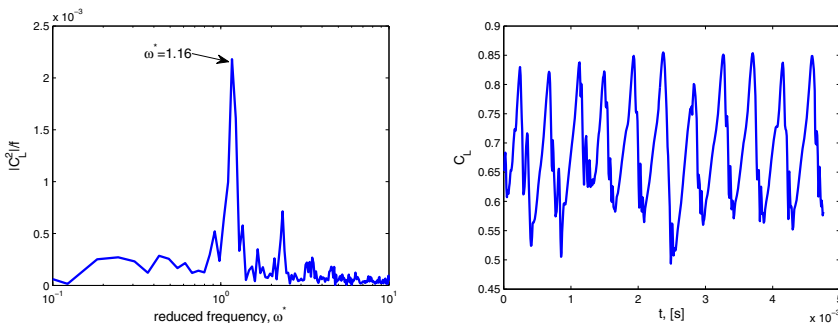


Fig. 14 Fourier transformed fluctuations of the lift coefficient C_L and corresponding fluctuations over time for a RANS simulation

In Fig. 14 the lift coefficient is investigated. The shock oscillation is highly periodic indicated by the pronounced peak in Fig. 14 (left). The reduced frequency shows a value of 1.16 which is about twice as high as it was expected from experiments at higher Reynolds numbers [28] and it is known that the influence of the Reynolds number on the reduced frequency is small [17]. The amplitude of the lift coefficient ΔC_L is about 0.3 (see Fig. 14, right). It is known from previous numerical investigations [26] that for the DRA2303 case the reduced frequencies of the RANS simulations were far off the expected values. However, numerical investigations with RANS on other airfoil types proved to be much more promising and more successful than for the DRA2303 case [22].

For this reason, a full LES simulation was set up to examine the physical aspects of buffet without using standard turbulence models. The simulation time for this configuration was about $40c/U_\infty$ which can be considered as a long term simulation where effects of initial perturbations or flow developing effects possess no influence anymore on the solution and a periodic flow behavior determines the result.

In Fig. 15 the instantaneous pressure coefficient c_p and the average pressure coefficient fluctuations at the upper side of the profile are presented. Note that the extension of the horizontal shock oscillation is about $0.07c$. That is much smaller compared to the outcome of the RANS simulation. The peak in the average pressure fluctuations is more pronounced (compare Fig. 13, right) and near the trailing edge the intensity of the fluctuations increases but they never exceed the strength of the shock. Upstream of the shock the pressure fluctuations are very small compared to the fluctuations at the shock position and downstream of the shock. The reduced frequency ω^* of the lift coefficient oscillation of the full LES solution presented in Fig. 16 is about 0.74 and thus much lower than the one obtained by the RANS simulation. Again, the oscillation is highly periodic which is indicated by the peak in the frequency spectrum. The amplitude of the lift coefficient $\Delta C_L \approx 0.03$ is small compared to that of the RANS simulation. This finding is due to the smaller horizontal shock oscillation amplitude and the less pronounced pressure fluctuations downstream of the shock.

Figure 17 compares the fluctuating pressure intensity at different locations. At the upper side at $0.25c$ the amplitude of the fluctuation is very low, however, a small distinctive bump is evident at $\omega^* \approx 0.7$. Near the shock at $0.55c$ a peak occurs at the reduced frequency $\omega^* = 0.74$ and the distribution of the values is very similar to that in Fig. 16. Close to the trailing edge at $0.9c$ the major peak is still at $\omega^* \approx 0.73$ but the pressure fluctuations at higher reduced frequencies have grown stronger compared to the position near the shock. This is due to the turbulent shear layer which is at this flow configuration maximum in size at the trailing edge. Although the pressure fluctuations are averaged in the spanwise direction the pressure fluctuations at very high reduced frequencies can be related to the turbulent shear layer. At the lower side of the profile at $0.9c$ the intensity level of the fluctuations is one order of magnitude smaller compared with the corresponding position at the upper side. However, a distinct peak at $\omega^* \approx 0.73$ is visible which is caused by the direct influence of the oscillating shock at the upper side.

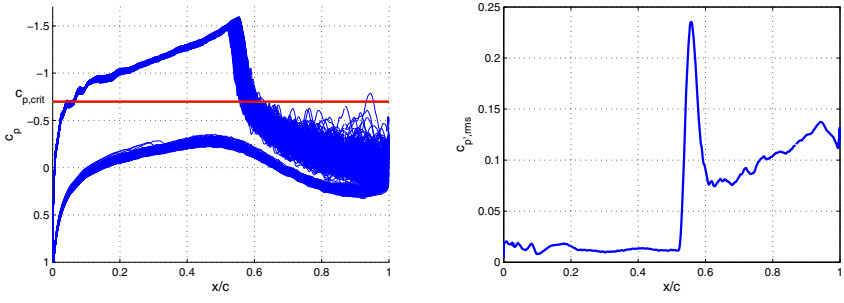


Fig. 15 Fluctuating wall pressure coefficient c_p (left) and corresponding *rms*-values at the upper side of the profile (right) for an LES simulation

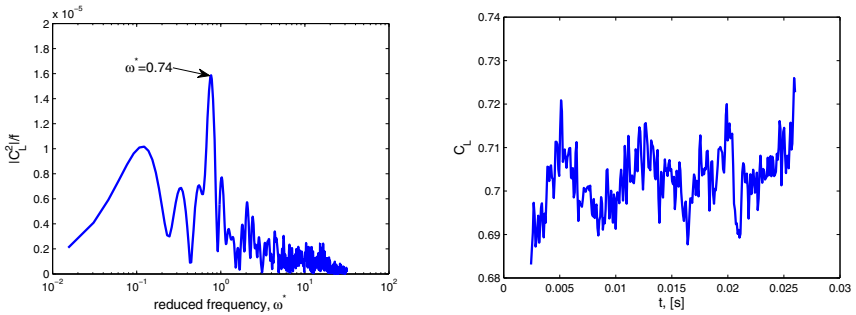


Fig. 16 Fourier-transformed fluctuations of lift coefficient C_L and corresponding fluctuations over time for an LES simulation

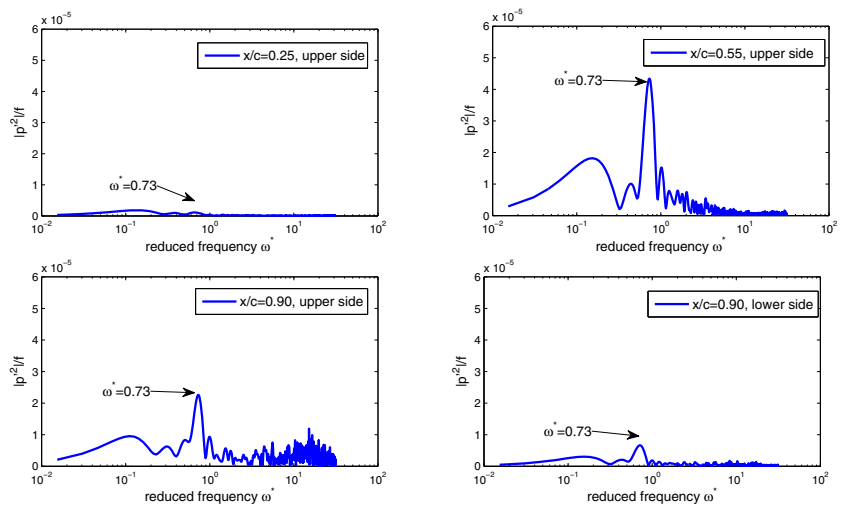


Fig. 17 Fourier transformed pressure fluctuations at different positions of the DRA2303-airfoil

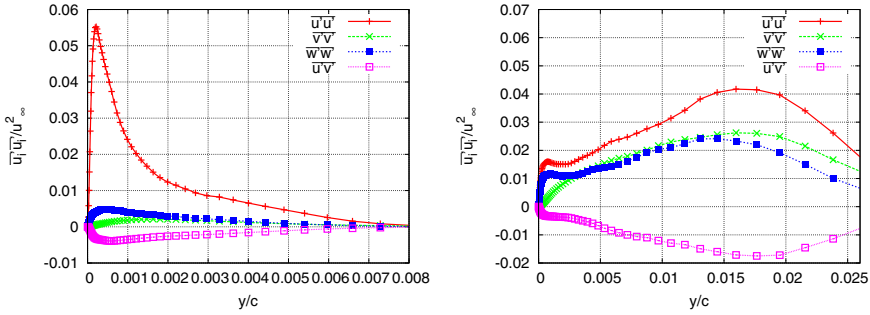


Fig. 18 Reynolds shear stresses at two different positions at the DRA2303-profile: at $x/c = 0.4$ (left) and $x/c = 0.6$

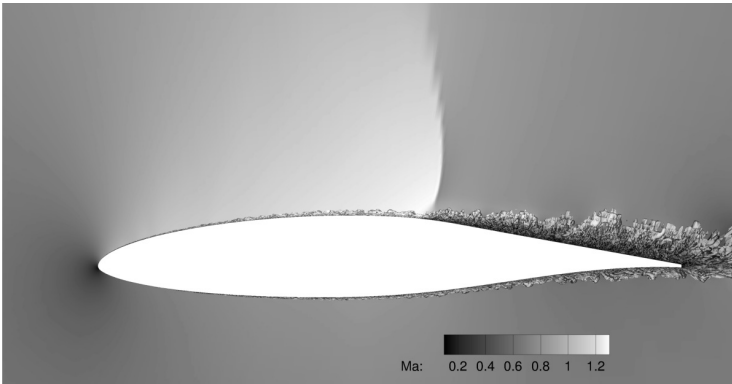


Fig. 19 λ_2 structures of transonic flow around a DRA2303 profile for a full domain LES simulation

The Reynolds shear stresses of the averaged full LES solution at two different locations is exhibited in Fig. 18. At $0.4 c$ the distribution resembles that of a flat plate boundary layer flow and its turbulent features. Behind the shock, however, the maxima of all components moves to the center of the boundary layer and the intensity level of $\langle v'v' \rangle$, $\langle w'w' \rangle$, and $\langle u'v' \rangle$ is much higher compared with the position ahead of the shock. Downstream of the shock the intensities of $\langle u'u' \rangle$, $\langle v'v' \rangle$, and $\langle w'w' \rangle$ are at the same level which indicates that the turbulent structures show an isotropic behavior compared with the high level of anisotropy of the near-wall turbulence upstream of the shock.

The λ_2 -contours [13] are shown in Fig. 19. After the interaction with the shock, the turbulent boundary layer separates and a shear flow is formed that develops large coherent structures that convect near the trailing edge. These large structures are responsible for the high level pressure oscillations at high frequencies near the trailing edge.

Due to the very different transient behavior regarding buffet and the subsequent consequences for the boundary layer downstream of the shock location the *rms*

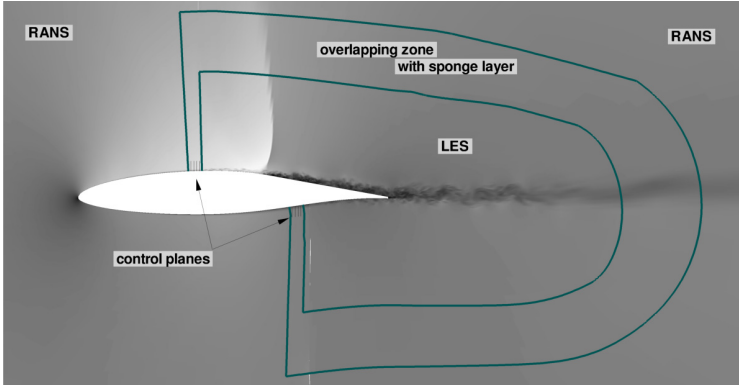


Fig. 20 Computational setup of a fully coupled zonal RANS-LES simulation

values of the pressure coefficient fluctuations diverge in the region between $0.4c$ and the trailing edge at the upper side of the airfoil (figs. 13 and 15). Because of the known drawbacks of RANS simulations (see section 1) a zonal RANS-LES computation was set up to overcome the turbulence modeling issue at the shock location and downstream of it. The extent of the LES domain was chosen like in the evaluation of the sensor results which have been discussed in section 3.1. The resulting computational set up is shown in Fig. 20. Due to the complexity of this particular case the results can be considered just preliminary.

At the inlet of the LES domain on the upper and lower side of the airfoil, the zonal SEM approach (see section 2.3) is used to generate synthetic turbulent structures. Downstream of the inlet four control planes are located between $0.37c$ and $0.4c$ at the upper side and between $0.7c$ and $0.73c$.

Since pressure waves, caused by the transient shock behavior, travel from the LES domain to the RANS domain and vice versa the time window where the solutions are averaged has to be carefully defined. On the one hand, the turbulent flow properties of the LES solution have to be averaged properly over a sufficiently large time window before being transferred to the RANS domain. On the other hand, the amplitude and frequency of the traveling pressure waves caused by the shock must be captured in a time window which is as small as possible to prevent a significant alteration of the pressure wave signal. A time window of the size of $1c/U_\infty$ was found to satisfy these requirements.

Table 8 Numerical details of the LES domain for the zonal RANS-LES simulation

	streamwise x	wall normal y	spanwise z
Grid points ($13.7 \cdot 10^6$)	1430	97	99
Resolution, wall units	$\Delta x^+ \approx 35$	$\Delta y_{min}^+ \approx 1.0$	$\Delta z^+ \approx 20$

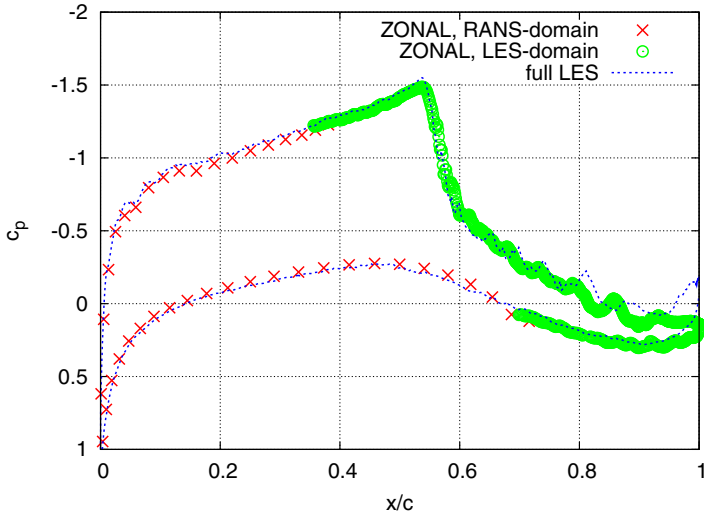


Fig. 21 Comparison of instantaneous wall pressure coefficient c_p between full domain LES and zonal RANS-LES computation

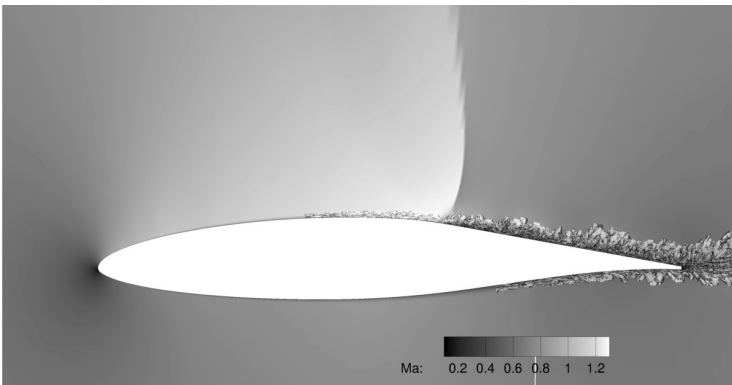


Fig. 22 λ_2 structures of transonic flow around a DRA2303 airfoil for a fully coupled zonal RANS-LES solution

In Fig. 21 the instantaneous pressure coefficients c_p are given. The zonal RANS-LES solution shows a good agreement with the full LES solution concerning shock position and strength. Downstream of the shock the c_p evolution shows minor discrepancies near the trailing edge. However, upstream of the shock the zonal RANS-LES results agree pretty well with the findings of the full LES. Note the smooth transition from RANS to LES of the pressure coefficient near the overlapping zones at approximately $0.37 c$ (upper side) and $0.7 c$ (lower side).

The λ_2 -contours of the instantaneous zonal RANS-LES solution are depicted in Fig. 22. They show the same features compared to the contours of the full LES

simulation such as large coherent structures downstream of the shock interaction zone convecting towards the trailing edge.

4 Conclusion

In this article, a sensor was presented which is capable of detecting flow regions of RANS simulations, where the intricacy of the flow field requires a higher order turbulence model. This sensor was applied to several flow cases with complex flow phenomena such as transonic airfoil flow with buffet, or shock wave turbulent boundary-layer interaction. For these specific cases the flow regions, where the structure of the flow is highly complex, were replaced by LES domains. To couple the RANS with the LES domain synthetic turbulence generating approaches and control planes were used which transferred the turbulent intensities of the RANS solution to the LES domain. Two different STGM were tested and validated for a subsonic zero pressure gradient boundary layer flow (see section 3). The zonal SEM approach was found to be superior compared to the zonal Batten approach and therefore used in subsequent computations.

Different configurations of control planes were tested on a supersonic zero pressure gradient boundary layer flow to evaluate the quality of the transition from the introduced synthetic turbulent structures to fully turbulent flow.

The case where an oblique shock impinges on a compressible turbulent boundary layer was numerically investigated and discussed. First, the sensor detected the required size of the required LES domain by evaluating the corresponding full domain RANS solution. Second, the RANS domain was coupled with the LES region by the zonal SEM approach and control planes downstream of the inlet. This constituted the transition from RANS to LES. The transition from the LES back to the RANS domain was done by applying the approach of König *et al.* [15]. The results have shown that the full domain LES and the zonal RANS-LES provided superior solutions compared to a full domain RANS simulation. The extension of the required LES domain in the zonal RANS-LES approach was well predicted by the sensor.

Finally, the computational results of the DRA2303 airfoil in a transonic flow configuration were thoroughly discussed. The results of a full domain RANS and full domain LES were compared and the sensor was applied to the RANS solution. The resulting LES domain extensions were implemented into a zonal grid. The coupled zonal RANS-LES approach provided preliminary results and indicates that the quality of the solution is comparable to that of the full domain LES solution.

These two presented cases have shown that the solution computed by a zonal RANS-LES simulation agree well with the corresponding full domain LES solutions. However, the computational costs due to the considerable reduction of grid points were reduced by a factor of 2.

References

- [1] Baldwin, B.S., Lomax, H.: Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. Technical Report 78-257, AIAA (January 1978)
- [2] Batten, P., Goldberg, U., Chakravarthy, S.: Interfacing statistical turbulence closures with large-eddy simulation. *AIAA Journal* 42(3), 485–492 (2004)
- [3] Boris, J.P., Grinstein, F.F., Oran, E.S., Kolbe, R.L.: New insights into large eddy simulation. *Fluid Dynamics Research* 10, 199–228 (1992)
- [4] Celić, A., Hirschel, E.H.: Comparison of eddy-viscosity turbulence models in flows with adverse pressure gradient. *AIAA Journal* 44(10), 2153–2169 (2006)
- [5] Driver, D.M., Johnston, J.P.: Experimental study of a three dimensional shear driven turbulent boundary layer with streamwise adverse pressure gradient. Technical report, NASA (1990)
- [6] Dussauge, J.P., Dupont, P., Debiève, J.F.: Unsteadiness in Shock Wave Boundary Layer Interaction with Separation. *Aerospace Science and Technology* 10, 10–75 (2006)
- [7] Dussauge, J.P., Piponnier, S.: Unsteadiness in Shock/boundary-layer interaction: Possible sources of unsteadiness. *Journal of Fluids and Structures* 10, 10–75 (2008)
- [8] El-Askary, W.A., Meinke, M., Schröder, W.: Large Eddy Simulation of a Compressible Boundary Layer. Pap., Aerodyn. Inst. RWTH Aachen. Presented at GAMM Conference 2002, Augsburg, March 25-28 (2002)
- [9] von Fröhlich, J., Terzi, D.: Hybrid les/rans methods for the simulation of turbulent flows. *Prog. Aerospace Sci.* 44, 349–377 (2008)
- [10] Fulker, J.L., Simmons, M.J.: An Experimental Investigation of Passive Shock/Boundary Layer Interaction Control on an Aerofoil. *Draiasihwaicr 9521611 Euroshock Tr. Aer.* 2, 4913(2) (1992)
- [11] Garnier, E., Sagaut, P., Deville, M.: Large Eddy Simulation of Shock/Boundary-Layer Interaction. *AIAA Journal* 40(10), 79–107 (2002)
- [12] Jarrin, N., Benhamadouche, S., Laurence, D., Prosser, R.: A synthetic-eddy-method for generating inflow conditions for large-eddy simulations. *International Journal of Heat and Fluid Flow* 27, 585–593 (2006)
- [13] Jeong, J., Hussain, F.: On the identification of a vortex. *J. Fluid Mech.* 285, 69–94 (1995)
- [14] Keating, A., de Prisco, G., Piomelli, U.: Interface conditions for hybrid rans/les calculation. *International Journal of Heat and Fluid Flow* 27, 777–788 (2006)
- [15] König, D., Schröder, W., Meinke, M.: Embedded les-to-rans boundary in zonal simulations. *Journal of Turbulence* 11, 1–25 (2010)
- [16] Kraichnan, R.H.: Inertial Ranges in Two-Dimensional Turbulence. *Physics of Fluids* 10(7), 1417–1423 (1967)
- [17] Lee, B.H.K.: Self-sustained shock oscillations on airfoils at transonic speeds. *Prog. Aerospace Sci.* 37, 147–196 (2000)
- [18] Liou, M.-S., Steffen, C.J.: A new flux splitting scheme. *Journal of Computational Physics* 107, 23–39 (1993)
- [19] Meinke, M., Schröder, W., Krause, E., Rister, T.: A comparison of second- and sixth-order methods for large-eddy simulations. *Computers and Fluids* 31, 695–718 (2002)
- [20] Menter, F.R.: Eddy viscosity transport equations and their relation to the $k - \epsilon$ -model. *Journal of Fluids Engineering* 119, 876–884 (1997)
- [21] Smirnov, A., Shi, S., Celik, I.: Random flow generation technique for large eddy simulations and particle dynamics modeling. *Journal of Fluids Engineering* 123, 359–371 (2001)

- [22] Soda, A.: Numerical Investigation of Unsteady Transonic Shock/Boundary-Layer Interaction for Aeronautical Applications. Technical report, DLR, Göttingen, Forschungsbericht (2007)
- [23] Spalart, P.R., Allmaras, S.R.: A One-Equation Turbulence Model for Arodynamic Flows. In: 30th Aerospace Sciences Meeting & Exhibit, Reno, January 6-9, AIAA Paper 92-0439 (1992)
- [24] Spille, A., Kaltenbach, H.-J.: Generation of turbulent inflow data with a prescribed shear-stress profile. In: Third AFSOR Conference on DNS and LES (August 2001)
- [25] Touber, E., Sandham, N.D.: Large-eddy simulation of low-frequency unsteadiness in a turbulent shock induced separation bubble. *Theor. Comput. Fluid Dyn.* 23, 79–107 (2009)
- [26] Wagner, W.: An Experimental Investigation of Passive Shock/Boundary Layer Interaction Control on an Airfoil: Unsteady Measurements. In: Hirschel, E.H. (ed.) EUROSHOCK. Drag Reduction by Passive Shock Control; Results of the Project EUROSHOCK. NNFM, vol. 36, pp. 401–414. Vieweg Verlag, Braunschweig (1997)
- [27] Wilcox, D.C.: *Turbulence Modeling for CFD*. DCW Industries (1993)
- [28] Wolles, B.A.: Computation of Transonic Flows Applying Shock Boundary Layer Interaction Control. In: Hirschel, E.H. (ed.) EUROSHOCK. Drag Reduction by Passive Shock Control; Results of the Project EUROSHOCK. NNFM, vol. 36, pp. 291–318. Vieweg Verlag, Braunschweig (1997)
- [29] Zhang, Q., Schröder, W., Meinke, M.: A zonal rans/les method to determine the flow over a high-lift configuration. Accepted for publication in *Computers and Fluids* (2010)

Part III
Numerical Methods

The Application of Iterated Defect Corrections Based on WENO Reconstruction

Alexander Filimon and Claus-Dieter Munz

Abstract. In this article we apply the procedure of the iterated defect correction method to the Euler equations as well as to the Navier-Stokes equations. One building block in the defect correction approach is the lower order basic method, usually first or second order accurate. This scheme gives a steady solution of low accuracy as the starting point. The second building block is the WENO reconstruction step to estimate the local defect. The local defect is put into the original equation as source on the right hand side with a minus sign. The resulting modified equation is then again solved with the low order scheme. Due to the source term with the local defect the order of accuracy is iteratively shifted to the order of the reconstruction. We show numerical results for several validation test cases and applications.

1 Introduction

Numerical simulations of the equations of fluid mechanics contain unavoidable errors due to several necessary approximations. To analyze these errors is crucial for the evaluation of the reliability of the numerical results. In the following we focus ourselves to the discretization errors. This means, that the modeling errors are excluded and the exact solution of the governing equations is supposed to be the reference solution of the described physical phenomenon.

The discretization errors can be separated into local and global discretization errors. By inserting the exact solution into the discretized equations, the local discretization error, also known as the local defect of a numerical approximation, can be determined. The more significant global discretization error gives the difference

Alexander Filimon · Claus-Dieter Munz
Institute of Aerodynamics and Gas Dynamics, 70569 Stuttgart, Germany
e-mail: filimon,munz@iag.uni-stuttgart.de

between the numerical and the exact solution. In both cases the exact solution is needed, which makes the error approximation for real applications cumbersome. A common approach is to run the same problem on several meshes with different step size h . A mesh convergence study allow then to compute the so called experimental convergence rate. Finally, a Richardson extrapolation can be used to determine the best approximate solution together with an estimation of the global discretization error. In practical 3D applications with complex geometries, this approach becomes cumbersome and sometimes even unfeasible because of the high computational costs. Our approach allows an error approximation for steady problems on the original mesh by using a polynomial reconstruction within the defect correction method.

Starting with a steady solution of a first or second order accurate finite volume scheme, we employ the modified weighted essentially non oscillatory (WENO) reconstruction scheme of Dumbser and Käser [4] for unstructured meshes. The resulting polynomial distribution allows an improved flux computation which can be applied to estimate the local discretization error. The method of the iterated defect correction (IDeC) consists of subtracting this local defect as a source term on the right hand side of the original equations [23, 19]. The now modified equations are solved with the original method of first or second order accuracy, in the following also called the basic method or the basic scheme, resulting into a new corrected steady solution. A further reconstruction of the corrected solution yields a better estimation of the local defect which is now used in the modified equations. Iteratively applied, the method of the defect correction shifts the order of accuracy of the basic scheme to the higher order of the used reconstruction. By this approach, an approximation of the global discretization error up to an accuracy of the higher order reconstruction is available.

2 The Method of Iterated Defect Correction

The defect correction approach was originally proposed by Zadunaisky ([22], [23]) for the estimation of the global discretization error in ordinary differential equations. The method was then generalized by Stetter [18] who introduced the iterative procedure which is applied in this work to partial differential equations. A number of theoretical investigations were done for ordinary differential equations by Frank ([5], [6]). Further papers of Pereyra ([15], [14]) show a different way of applying the defect correction method and gives additional analysis of the method. A nice overview on the defect correction approach can be found in Stetter [19] where he gives an overview of the different procedures. The proposed iterative defect correction method in this work is based on the procedure introduced by Stetter in [18] for ordinary differential equations. As was shown by Frank and Ueberhuber in [7] the iterated defect correction can also be applied on partial differential equations.

For the sake of simplicity we describe in the following the employed iterated defect correction on the example of a scalar one dimensional evolution equation

$$u_t + f(u)_x = 0, \quad (1)$$

keeping in mind that the whole procedure can be extended to multi dimensions and to diffusion fluxes which additionally depend on ∇u , as it is the case for the Navier-Stokes equations. As mentioned before, we focus on steady solutions, i.e., $u_t = 0$. The time dependence in (1) is used only for the iteration of the approximate solution to a steady state. Conform with the convention in the cited papers we write the equation (1) in the abstract form

$$L_h u_h = r_h \quad \text{with} \quad Lu \equiv \frac{\partial f}{\partial x}, \quad (2)$$

where the operator L is the exact linear or nonlinear differential operator and L_h is the discretized operator with the mesh width parameter h . For L_h we impose a stable, consistent and fast invertible operator, which is easily achieved by an operator with a low consistency order of one or two. For the theory of the iterated defect correction the operator L_h can also be of higher order. For practical calculations it is more interesting to correct a first or second order accurate method which is often applied in practice. Equation (2) will be called the basis method with the solution u_h computed by inversion of the operator L_h :

$$u_h = L_h^{-1} r_h. \quad (3)$$

Additionally we need another numerical method for the original problem (1) on the same mesh, but with a higher consistency order

$$S_h w_h = r_h. \quad (4)$$

(4). This higher order discretization will only be used to estimate the local defect and is applied once per defect iteration in that form. Instead of solving directly the higher order discrete problem which may need a lot of time and development of the solution procedure, the modified problem

$$L_h u_h = r_h - d_h \quad (5)$$

is solved using the basic method (2). With $d_h = S_h w_h - L_h u_h$ the local defect we apply equation (5) iteratively

$$L_h u_h^{[k+1]} = r_h - d_h^{[k+1]} \quad k = 1, 2, 3, \dots, N_{IDeC} \quad (6)$$

with the defect iteration index k and N_{IDeC} the maximum defect correction iterations, converging towards the solution w_h , the reconstruction polynomial higher order. The whole defect correction procedure can be formulated in the following step by step description.

1. We start with a steady solution

$$u_h^{[0]} = L_h^{-1} r_h^{[0]} \quad \text{with} \quad r_h^{[0]} = r_h \quad (7)$$

of our basic method (2). The approximated solution u_h is then reconstructed with the weighted essentially non oscillatory scheme, which will be described in section 3. The reconstruction produces a polynomial distribution $w_h^{[k]}$ of the integral average in each grid cell.

2. Applying now the operator of higher order consistency S_h to the reconstructed solution $w_h^{[k]}$ we compute the local defect $d_h^{[k+1]}$ for the next defect correction iteration $k+1$:

$$\begin{aligned} d_h^{[k+1]} &= S_h w_h^{[k]} - r_h^{[k]} = S_h w_h^{[k]} - L_h u_h^{[k]}, \\ r_h^{[k]} &= r_h - d_h^{[k]}. \end{aligned} \quad (8)$$

3. This defect is then subtracted as a source term on the right hand side of equation (2) and the modified equation

$$\begin{aligned} L_h u_h^{[k+1]} &= r_h - d_h^{[k+1]} \\ &= r_h - \left(S_h w_h^{[k]} - L_h u_h^{[k]} \right) \end{aligned} \quad (9)$$

is solved with the basic method, applying the inverse operator L_h^{-1} . One gets then the corrected solution $u_h^{[k+1]}$ after the k -th defect iteration, converging for

$$L_h u_h^{[k+1]} \approx L_h u_h^{[k]}. \quad (10)$$

With this convergence criteria, equation (9) reduces to the method higher order (4), with w_h the reconstruction polynomial.

Equation (5) is also called the "neighboring problem" with respect to the original formulation of the defect correction by Zadunaisky. For the process of iterated defect correction one must assume that (6) and (2) are neighboring mathematical problems. Since w_h represents a polynomial reconstruction of the basic solution u_h on the same mesh, the defect d_h as defined in equation (8) is "small" and so the assumption of the "neighboring problem" is plausible.

As we use a finite volume scheme for the basic method we can write the modified equation (5) in a semi discrete form defined on the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [t^n, t^n + \Delta t]$ with $u_i = u_h$, being the discrete state in the cell i , as

$$\Delta u_i^{[k+1]} = -\frac{1}{\Delta x} R(u_i^{[k+1]}) - \frac{1}{\Delta x} \underbrace{\left(R(w_i^{[k]}) - R(u_i^{[k]}) \right)}_{d_h^{[k+1]}} \quad (11)$$

with

$$R(\mathcal{Q}_i) = \int_{t^n}^{t^n + \Delta t} \int_{x_{i-1/2}}^{x_{i+1/2}} f(\mathcal{Q}_i)_x dx dt, \quad (12)$$

whereas \mathcal{Q}_i is considered as a placeholder for $u_i^{[k]}$, $u_i^{[k+1]}$ and $w_i^{[k]}$. As common for a finite volume approach the unknown physical flux $f(\mathcal{Q}_i)$ is replaced by an appropriate numerical flux approximation $g_{i+\frac{1}{2}}$ on the cell border of two adjacent cells. The numerical flux depends only on the states left and right of the cell interface: $g_{i+\frac{1}{2}} = g(\mathcal{Q}_{i+\frac{1}{2}}^-, \mathcal{Q}_{i+\frac{1}{2}}^+)$, with \mathcal{Q}^- defining the state on the interface in the cell itself and \mathcal{Q}^+ being the state on the interface in the neighboring cell. If we choose as a simple example a basic method of first order in time and space, equation (11) yields

$$\Delta u_i^{[k+1]} = -\frac{\Delta t}{\Delta x} \left(g(u_{i+\frac{1}{2}}^{[k+1,-]}, u_{i+\frac{1}{2}}^{[k+1,+]}) - g(u_{i-\frac{1}{2}}^{[k+1,-]}, u_{i-\frac{1}{2}}^{[k+1,+]}) \right) - d_h^{[k+1]} \quad (13)$$

with

$$d_h^{[k+1]} = \frac{\Delta t}{\Delta x} \left(g(w_{i+\frac{1}{2}}^{[k,-]}, w_{i+\frac{1}{2}}^{[k,+]}) - g(w_{i-\frac{1}{2}}^{[k,-]}, w_{i-\frac{1}{2}}^{[k,+]}) \right) - \frac{\Delta t}{\Delta x} \left(g(u_{i+\frac{1}{2}}^{[k,-]}, u_{i+\frac{1}{2}}^{[k,+]}) - g(u_{i-\frac{1}{2}}^{[k,-]}, u_{i-\frac{1}{2}}^{[k,+]}) \right) \quad (14)$$

an approximation of the corrected state $u_h^{[k+1]}$ in the cell i . We point out that the integral of the higher order fluxes $f(w_h^{[k]})$ have to be computed with an appropriate numerical integration of accuracy higher than the one of the basic method. In the case of 1D there is no need of such an integration, the interface being the only integration point. For 2D or 3D discretizations an efficient integration scheme is necessary. In our case we use Gauss quadrature. This leads to an approximation of the flux integral in cell i as

$$\int_{\partial C_i} g(w_i^{[k,-]}, w_i^{[k,+]}) \mathbf{n} dS \approx \sum_{K \in \partial C_i} \sum_{j=1}^{n_{GP}} \omega_j^K g(w_{i,j}^{[k,+]}, w_{i,j}^{[k,-]}) \mathbf{n}_K S_K, \quad (15)$$

where n_{GP} and ω_j^K denote the number and the weights of the Gauss integration points j on the interface K , respectively, S_K is the length or the surface and \mathbf{n}_K is the outward pointing unit normal vector. If the polynomial degree of the reconstruction is chosen to be $p' > p$, with p , being the polynomial degree of the basic method, we take $n_{GP} = \frac{p'+1}{2}$ for the 2D case, which is accurate up to a polynomial degree of p' . In the 3D case we use a rather sub-optimal number $n_{GP} = (\frac{p'+1}{2})^2$ of Gauss points which is nevertheless accurate up to a polynomial degree p' .

2.1 IDEC for Inhomogeneous Problems

For inhomogeneous equations the method of iterated defect correction can be applied in two different ways, which leads both to the same corrected solution but with different time efficiencies. The difference is even bigger for source terms depending on the solution itself. In the following work we will present both inhomogeneous equations with source terms depending only on space and source terms including the solution itself. To describe the different formulations we take the scalar evolution equation

$$u_t + f(u)_x = s(u). \quad (16)$$

again in 1D as example with the source term s depending on the solution u . If we apply the iterated defect correction on this problem as done before, computing the local defect only in the flux terms, we can write the modified equation (5) as

$$u_i^{[k+1]} + f(u^{[k+1]})_x = s(w) - \underbrace{\left(f(w^{[k]})_x - f(u^{[k]})_x \right)}_{d^{[k+1]}}. \quad (17)$$

The integration of each term is done as shown above what leads to a similar semi discrete representation of the modified equation (5) with the additional integral

$$\int_{t^n}^{t^n + \Delta t} \int_{x_{i-1/2}}^{x_{i+1/2}} s(w_i) dx dt,$$

of the source term s in the cell i . To achieve the consistency order of the reconstruction in the iterated defect correction procedure with the above formulation (17), it is important to compute the source term with the high order accuracy. This implies a reconstruction in each iteration of the basic method and in 2D and 3D an integration with much more Gauss points than used for the basic scheme of lower order is necessary. The high computational cost can be reduced by reformulating the problem in equation (17). Instead of taking only the fluxes into account for the local defect, we propose to include the source term as well in the definition of the local defect. This yields

$$u_i^{[k+1]} + f(u^{[k+1]})_x = s(u^{[k+1]}) - \underbrace{\left[f(w^{[k]})_x - s(w^{[k]}) - \left(f(u_h^{[k]})_x - s(u^{[k]}) \right) \right]}_{d^{[k+1]}}, \quad (18)$$

a new modified equation where the source term in the iteration of the basic scheme is now integrated with the lower order accuracy whereas the reconstruction of the source term is only done once per defect iteration to compute the local defect.

3 WENO Reconstruction on Unstructured Grids

In order to compute the higher order operator for the grid cell $C_{(i)}$ a reconstruction of the cell averages defined

$$\bar{u}_{(i)} = \frac{1}{|C_{(i)}|} \int_{C_{(i)}} u dV \quad (19)$$

is done, where $|C_{(i)}|$ is the length, the surface or the volume of the grid cell depending on the space dimension. To ensure a stable reconstruction even at discontinuities, a high order Weighted Essentially Non Oscillatory (WENO) reconstruction was chosen. This method was first introduced by Shu et al. [11, 10] and Osher et al. [13]. For the proposed defect correction method, the modified reconstruction algorithm by Dumbser et al. [4] is used which ensures a robust method on 2D and 3D unstructured meshes even with distorted cells [4] eliminating scaling and bad conditioning problems common to WENO reconstruction technique.

In this approach, the reconstructed polynomial is built by a linear combination of orthogonal basis functions as given in [3] for triangles in 2D and tetrahedrons in 3D. We write the reconstruction polynomial for the element $C_{(i)}$ as

$$w_{(i)}(\xi, \eta, \zeta) = \sum_{l=1}^L \hat{w}_{(i)_l} \Psi_l(\xi, \eta, \zeta), \quad (20)$$

with ξ, η and ζ the coordinates in the reference coordinate system. Unlike the common WENO reconstructions at discrete cell points, the basis polynomials are continuously extended over the whole stencil and are then restricted to the considered element $C_{(i)}$ after having obtained a reconstruction polynomial. The number of degrees of freedom L being $L = \frac{1}{2}(M+1)(M+2)$ in 2D and $L = \frac{1}{6}(M+1)(M+2)(M+3)$ in 3D depends on the polynomial degree M of the basis functions Ψ_l . Whereas the basis functions are space dependent, the reconstructed degrees of freedom $\hat{w}_{(i)_l}$ depend only on time.

Similar to the finite element framework, the reference space is the unit element C_U . This is a triangle with the canonical coordinates $(0, 0)$, $(0, 1)$ and $(1, 0)$ in 2D and a tetrahedron with the canonical coordinates $(0, 0, 0)$, $(0, 0, 1)$, $(0, 1, 0)$ and $(1, 0, 0)$ in 3D. The transformation from the physical space $x - y - z$ into the reference space $\xi - \eta - \zeta$ can be done by a linear transformation matrix (see [4]). To perform the reconstruction, several stencils have to be chosen which is done in the reference space. There are three groups which are defined as follows:

1. First the central stencil is built by adding successively Neumann neighbors, i.e. immediate side neighbors of the considered cell $C_{(i)}$, to the stencil until the desired number of cells n_e in one stencil is reached. The size of n_e will be discussed later on in this chapter.

2. The following stencils, three of them in 2D and four in 3D, are chosen out of the primary sectors. As mentioned before this is done in the reference space ξ, η, ζ . So the primary sectors are spanned by the vectors starting from each vertex of the unit element C_U along the edges intersecting this vertex. Transformed elements are then successively added to the stencils.
3. As shown by Käser and Iske [12] it is favorable to take one more family of stencils into account than the two mentioned above. Although this increases the computational effort it ensures a stable and robust reconstruction in 2D and 3D configurations for special locations of the discontinuities. Additionally it improves the one sided reconstruction, e.g. at walls. The so-called reverse sectors are spanned by the negative vectors of each primary sector defined above.

This sums up to $n_s = 7$ and $n_s = 9$ stencils in 2D and 3D which are used for the reconstruction. At the domain boundaries or for the case that not all stencils could be filled up due to geometrical reasons, the total number of stencils n_s can decrease.

For a conservative reconstruction one must assure that the polynomial distribution w_i in each cell $C_{(i)}$ of the stencil m conserves the integral mean value of the cell at hand $C_{(i_s)}$.

$$\frac{1}{|C_{(i_s)}|} \int_{C_{(i_s)}} w_{(i)}(\xi) dV = \bar{u}_{(i_s)} \quad (21)$$

The evaluation of the conservation condition is carried out in the reference space. This is done by applying linear transformation matrix with respect to the element $C^{(i)}$ to each cell in the stencil, where the transformed elements are in the following denoted by $\tilde{C}_{(i)}$. Taking into account that the degrees of freedom $\hat{w}_{(i)}$ are not space dependent, the above equation results in

$$|J| \sum_{l=1}^L \left(\int_{\tilde{C}_{(i_s)}} \Psi_l(\xi) d\xi d\eta d\zeta \right) \hat{w}_{(i)_l} = |J| |C_{(i_s)}| \bar{u}_{(i_s)}. \quad (22)$$

The Jacobian determinant which is introduced due to the transformation appears on both sides of equation (22), so it cancels out and with it scaling effects are eliminated as well. Furthermore, Abgrall reports in [1] that ill-conditioned reconstruction matrices are also avoided through this effect.

As the transformation to canonical coordinates is only done for the reconstructed cell, the integration in equation (22) turns out to be non-trivial. This is not the case if a second transformation with respect to the reconstructed cell is done additionally. For more details see [4]. With it, the left hand side of equation (22) can again be easily integrated using the Gaussian quadrature with an appropriate accuracy. This yields the following linear system which have to be solved for the reconstructed degrees of freedom.

$$\underline{\underline{A}} \hat{\mathbf{w}} = \bar{\mathbf{u}} \quad (23)$$

For a number $n_e = L$ of elements per stencil the matrix $\underline{\underline{A}}$ becomes square and easy invertible, but for realistic meshes this leads to an unstable scheme. So, to ensure the

robustness of the reconstruction we enlarge the stencils, see also [12]. The number of the elements per stencil is chosen as $n_e = 1.5L$ in 2D and $n_e = 2L$ in 3D. In addition the matrix can contain linear dependent rows due to geometrical reasons. This means that the reconstruction matrix \underline{A} may not be invertible. This is avoided by adding successively new elements to the stencil if one of the singular values of the matrix becomes zero. The overdetermined system (23) can be solved by an algorithm of singular value decomposition or, as it is done in our framework, by a least-squares method with the constraint (21).

The degrees of freedom $\hat{w}_{(i)}$ are now known, so the polynomials $w_{(i)}(\xi, \eta, \zeta)$ on each stencil are known and the final nonlinear reconstruction polynomial $w_{(i)}^{WENO}$ in the cell $C_{(i)}$ of degree M is defined by

$$w_{(i)}^{WENO}(\xi, \eta, \zeta) = \sum_{s=1}^{n_s} \omega_s w_{(i)_s}(\xi, \eta, \zeta) = \sum_{s=1}^{n_s} \sum_{l=1}^L \omega_s \hat{w}_{(i)_s, l} \Psi_l(\xi, \eta, \zeta). \quad (24)$$

Unlike the common ENO (Essentially Non Oscillatory) schemes, where only the less oscillating polynomial is chosen, all reconstruction polynomials on each stencil are taken into account by a linear combination as done in eq. (24) with the normalized nonlinear weights ω_s

$$\omega_s = \frac{\tilde{\omega}_s}{\sum_{r=1}^{n_s} \tilde{\omega}_r} \quad \text{with} \quad \tilde{\omega}_s = \frac{\lambda_s}{(\varepsilon + \sigma_s)^r} \quad (25)$$

according to [11, 17, 4], whereas the non-normalized nonlinear weights $\tilde{\omega}_s$ depend on the linear weights λ_s and the oscillation indicators σ_s .

The parameters ε and r are set in the common range given in the literature [17, 4], i.e. $\varepsilon = 10^{-5} - 10^{-14}$ and $r = 2 - 8$. Thereby ε is regarded as a threshold for a division by zero which does not influence much the stability of the reconstruction scheme. The parameter r states the sensitivity of the nonlinear weights relative to the oscillation indicators σ_s . For bigger r the reconstruction procedure tends to an ENO behavior, whereas for smaller values the scheme becomes more oscillatory.

For the weights ω_s in (24) suitable oscillation indicators are necessary to ensure a robust reconstruction. In literature ([10, 12]) this is usually achieved by a scaling with the cell volume. As the reconstruction procedure is done in the reference coordinate system this is not necessary any more. Due to the definition of the polynomials (20) σ_s can furthermore be computed in a mesh independent way

$$\sigma_s = (\hat{\mathbf{w}}_s)^T \underline{\underline{\Sigma}} \hat{\mathbf{w}}_s \quad (26)$$

with $\hat{\mathbf{w}}_s$, the vector of the degrees of freedom of the polynomial on the stencil m and $\underline{\underline{\Sigma}}$ the universal oscillation matrix defined by

$$\Sigma_{lk} = \sum_{r=1}^M \sum_{\alpha=0}^r \sum_{\beta=0}^{r-\alpha} \int_{C_U} \frac{\partial^r}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} \Psi_l(\xi, \eta, \zeta) \cdot \frac{\partial^r}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} \Psi_k(\xi, \eta, \zeta) d\xi d\eta d\zeta, \quad (27)$$

whereas $\gamma = r - \alpha - \beta$. As the reconstruction basis functions Ψ are generally given, the oscillation matrix is neither dependent on the mesh, nor on the problem, i.e. it can be computed and stored in advance of a computation, considerably increasing the efficiency of this reconstruction method.

In contrast to the common WENO schemes the linear weights are not used for the improvement of the accuracy as was shown by Liu, Osher and Chan [13] but simply defined by

$$\lambda_s = \begin{cases} \lambda_c & \text{if } s = 1, \text{ i.e. } c \text{ is the index of the central stencil,} \\ 1 & \text{else} \end{cases} \quad (28)$$

according to Dumbser et al. [4], with $\lambda_c \gg 1$, which puts a high emphasis on the central stencil. It was shown in [4] that a choice of $\lambda_c = 10^2 - 10^5$ does not show sensitivities in the results. Nevertheless, lower λ_c yield better results at discontinuities and larger weights are favorable for smooth solutions.

4 Numerical Results

For the validation of the implemented iterated defect correction method exhaustive studies have been made for 1D, 2D and 3D Euler and Navier-Stokes problems. All simulations have been carried out with a standard finite volume scheme using ghost cells to impose boundary conditions. Depending on the test case a first or a second order basic scheme was used, whereas for the defect correction a polynomial WENO reconstruction up to 6th order was applied with the standard choice for the reconstruction parameters shown in section 3. In this section we first show some of the convergence studies of those validation cases and we end up with application test cases.

4.1 Convergence Studies

To validate our proposed iterated defect correction method for inhomogeneous problems we took as a first example in 1D the nonlinear Euler equations where a source term $\mathbf{s}(\mathbf{u}, A)$, depending on the solution \mathbf{u} and a given geometry A , was added. The equations in (29) are derived from the homogeneous Euler equations in three dimensions with the assumption of a continuous area variation (see also [20]).

$$\mathbf{u}_t + \mathbf{F}(\mathbf{u})_x = \mathbf{s}(\mathbf{u}) \quad \text{with} \quad \mathbf{s} = -\frac{1}{A} \begin{pmatrix} \rho u A_x \\ \rho u^2 A_x \\ u(e+p)A_x \end{pmatrix} \quad (29)$$

This gives an approximation of a 2D axi-symmetric nozzle flow with the x -axis as nozzle symmetry and $A(x)$ as cross-sectional area along the nozzle. In our case we took a smooth sinus function

$$A(x) = \begin{cases} 2, & -1 \leq x \leq -\frac{1}{2} \\ 2 - \sin^4\left(\pi\left(x + \frac{1}{2}\right)\right), & -\frac{1}{2} < x < \frac{1}{2} \\ 2, & \frac{1}{2} \leq x \leq 1 \end{cases} \quad (30)$$

for the cross-sectional area which is illustrated in the upper left corner of Fig. 1. We took a subsonic expansion with inflow and outflow pressure $p = 1$ and an inflow velocity $u = 0.2$ with an inflow mass flow $\rho u = 0.28$. We obtain then an inflow Mach number of $Ma = 0.2$ which can be introduced into the 1D nozzle theory to evaluate the exact state at the inflow and outflow section. This is imposed during the simulations which result into a symmetrical distribution of the state variables (see Fig. 1). We can clearly see the difference between the first order basic method and the corrected solution in both amplitude and location of the peak which is expected to be in the nozzle throat at $x = 0$. In this case a cubic polynomial reconstruction was chosen to compute the local defect.

To measure the exact error between the approximated solution \mathbf{u}_h and the exact solution \mathbf{u}_e we use continuous L^p norms

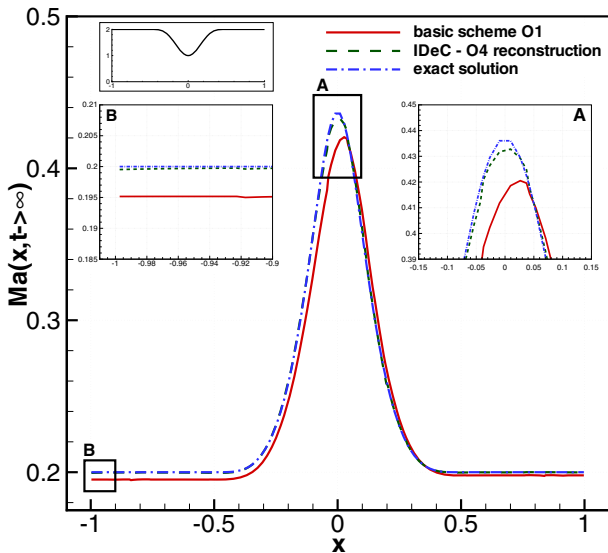


Fig. 1 Mach number distribution of a axi-symmetric nozzle flow simulated with a first order basic method and corrected by a fourth order polynomial reconstruction for the local defect

$$\| \mathbf{u}_h - \mathbf{u}_e \|_{L^p(\Omega)} = \left(\int_{\Omega} |\mathbf{u}_h - \mathbf{u}_e|^p dV \right)^{1/p}, \quad (31)$$

whereas the L^∞ -norm gives the maximum absolute error arising in the whole domain. We compute the integral with the Gaussian quadrature algorithm with twice the number of Gauss points compared to the numerical scheme. For the defect correction it is important to make this analysis with the high order polynomials and, respectively, with the high order integration and not with the accuracy of the basic method. Table I shows the convergence tables of the nozzle test case for five successively refined grids. In addition to the fourth order defect reconstruction we show a convergence table for a first order basic method corrected with a 6th order reconstruction to determine the local defect.

The validation of the 2D and 3D implementation has been carried out by using the method of manufactured solutions, i.e. we insert an analytical function

$$\rho_e(x) = \sin(\pi x) \cdot \sin(\pi y) + 2$$

for the exact density distribution in 2D and respectively

$$\rho_e(x) = \sin(\pi x) \cdot \sin(\pi y) \cdot \sin(\pi z) + 1$$

in 3D into the Euler equations and solve the inhomogeneous two-, respectively three-dimensional Euler equations

$$\mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{s}(\mathbf{x}). \quad (32)$$

Table 1 Iterated defect correction based on a first order steady solution with a 4th (up) and 6th (down) order reconstruction, 1D homogeneous grid, convergence rates of the mass flow variable

h	L^∞	L^1	L^2	\mathcal{O}_{L^∞}	\mathcal{O}_{L^1}	\mathcal{O}_{L^2}
Basic method $\mathcal{O}1 \rightarrow$ IDeC with $\mathcal{O}4$ reconstruction						
0.080	7.69E-03	2.17E-03	2.64E-03	-	-	-
0.040	6.34E-04	1.26E-04	1.78E-04	3.6	4.1	3.9
0.020	3.46E-05	6.08E-05	9.69E-05	4.2	4.4	4.2
0.010	2.14E-06	3.19E-06	5.63E-06	4.0	4.3	4.1
0.005	1.35E-07	1.83E-07	3.44E-07	4.0	4.1	4.0
Basic method $\mathcal{O}1 \rightarrow$ IDeC with $\mathcal{O}6$ reconstruction						
0.100	8.80E-03	2.22E-03	2.49E-03	-	-	-
0.067	1.92E-03	2.95E-04	4.19E-04	3.8	5.0	4.4
0.033	6.66E-05	8.94E-06	1.54E-05	4.9	5.0	4.8
0.017	1.54E-06	1.80E-07	3.28E-07	5.4	5.6	5.6
0.008	2.63E-08	3.02E-09	5.58E-09	5.9	5.9	5.9

The remaining state variables like the velocity and the pressure were set to a constant value greater zero. For our choice we obtain source terms which in contrast to the one dimensional analysis depend only on the space

$$s_i = \pi \cdot \sin(\pi x) \cdot \cos(\pi y) + \pi \cdot \cos(\pi x) \cdot \sin(\pi y) \quad \text{for} \quad i = 1 \dots 4 \quad (33)$$

in two dimensions and respectively

$$\begin{aligned} s_i &= \pi \cdot \sin(\pi x) \cdot \sin(\pi y) \cdot \cos(\pi z) + \pi \cdot \sin(\pi x) \cdot \cos(\pi y) \cdot \sin(\pi z) \\ &\quad + \pi \cdot \cos(\pi x) \cdot \sin(\pi y) \cdot \sin(\pi z) \quad \text{for} \quad i = 1 \dots 4 \\ s_5 &= \frac{3\pi}{2} \cdot \sin(\pi x) \cdot \sin(\pi y) \cdot \cos(\pi z) + \frac{3\pi}{2} \cdot \sin(\pi x) \cdot \cos(\pi y) \cdot \sin(\pi z) \\ &\quad + \frac{3\pi}{2} \cdot \cos(\pi x) \cdot \sin(\pi y) \cdot \sin(\pi z) \end{aligned} \quad (34)$$

in three dimensions. For the simulations we initialized the domain $\Omega_{2D} = [0; 1] \times [0; 1]$ in 2D and respectively $\Omega_{3D} = [0; 1]^3$ in 3D with the exact solution and iterated the basic scheme to a steady state with the exact solution imposed on the boundaries. The same convergence study as in one dimension based on the L^p -norms was carried out on four successively adapted grids. In all our computations we used fully unstructured grids with irregular triangles in two dimensions and tetrahedrons in three dimensions. Each adaptation is performed globally, i.e. we applied the so-called red-refinement in each cell of the domain per adaptation step. An example of two adaptation steps is shown for the three dimensional case in Fig. 2.

Again we can see the difference between the solution of the basic method and the corrected one. This is demonstrated for the two dimensional case for the density distribution in Fig. 3. We have to mention that for the visualization of the higher order solutions we subdivide the numerical grid and write out the value of the polynomial distribution at each barycenter center of the subdivided grid. In this way we can see the Godunov approach of constant values in the cell for the first order solution (Fig. 3 left) and the continuous fourth order solution with vanishing jumps between the cells (Fig. 3 right). A quantitative analysis can be done by determining the convergence rates for the corrected solutions as it was done in the one dimensional case. In Tab. 2 we show the experimental convergence order for the two- and

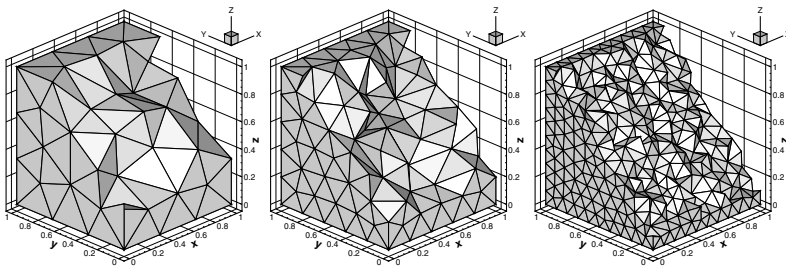


Fig. 2 Grid topologies used for the convergence studies in three dimensions

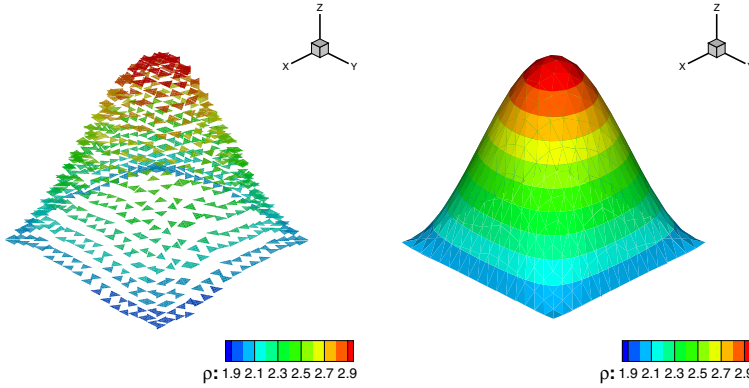


Fig. 3 Solution of the first order basic scheme (left) and the corrected solution of an IDEc with a 4th order polynomial reconstruction for the local defect (right)

three dimensional simulations where we used a first order basic scheme with a cubic polynomial reconstruction for the defect correction. In both cases we reach the optimal theoretical convergence order of $M + 1$ when iterated defect correction is applied. The convergence rates can approve that with the mesh step size $h \rightarrow 0$ the error tends towards zero with the potential power of $M + 1$, M being the polynomial degree of the reconstruction basis polynomials Ψ (see chapter 3). However, nothing can be said about the real simulation time needed by the employed numerical scheme and with it the real gain of using a higher order reconstruction. Therefore we show in Fig. 4 the L^1 -norm of the errors for the one and two dimensional computations described above over the total CPU-time in seconds needed

Table 2 Density convergence rates for IDEc with a first order basic scheme and a cubic polynomial reconstruction on 2D (up) and 3D (down) irregular unstructured grids

h	L^∞	L^1	L^2	$\hat{\sigma}_{L^\infty}$	$\hat{\sigma}_{L^1}$	$\hat{\sigma}_{L^2}$
<u>2D Basic scheme $\mathcal{O}1 \rightarrow$ IDEc with $\mathcal{O}4$ reconstruction</u>						
0.191	7.55E-02	1.06E-02	1.45E-02	-	-	-
0.096	5.78E-03	4.68E-04	7.12E-04	3.7	4.5	4.4
0.048	7.07E-04	2.95E-05	4.91E-05	3.0	4.0	3.9
0.024	4.97E-05	1.69E-06	3.17E-06	3.8	4.1	3.9
<u>3D Basic scheme $\mathcal{O}1 \rightarrow$ IDEc with $\mathcal{O}4$ reconstruction</u>						
0.182	5.67E-02	9.49E-03	1.22E-02	-	-	-
0.127	9.69E-03	1.87E-03	2.39E-03	5.0	4.6	4.6
0.068	8.71E-04	1.01E-04	1.32E-04	3.8	4.6	4.6
0.035	8.38E-05	7.09E-06	9.54E-06	3.6	4.1	4.0

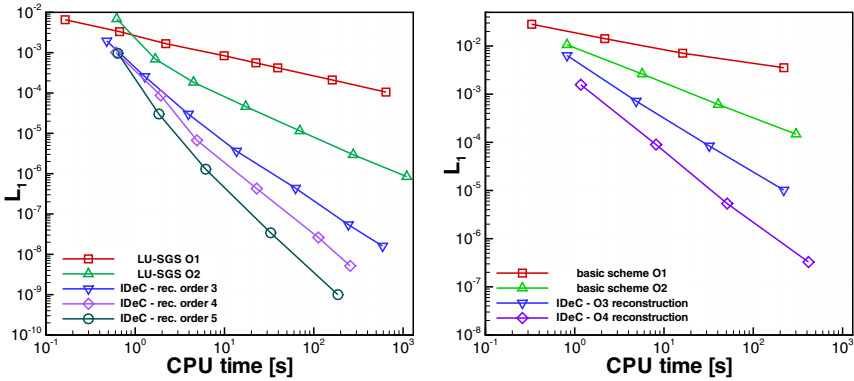


Fig. 4 Convergence rates over total CPU time for first and second order methods compared to iterated defect correction with higher order reconstruction in 1D (left) and 2D (right)

for a converged iterated defect correction and the basic schemes of first and second order. The simulations were all performed on one single AMD Athlon 5200+ processor with 3GB of RAM. So we can compare directly the computational effort to reach the same given accuracy of the L^1 -norm. To give an example, if we want to reach an error norm of $L^1 = 10^{-4}$ in the one dimensional case (Fig 4, left), we obtain a speed up of factor 4 comparing a second order scheme with a third order corrected solution based on a first order basic scheme. The same comparison for the two dimensional case leads even to a speed up of factor 20 which is due to higher computational cost for the 2D simulations concerning for example the integration (Fig 4, right). This results for the speed up are surely dependent on the test case, nevertheless they give an idea of the potential of higher order schemes. However, in the one dimensional test case we can see, that the speed up of a higher order scheme starts to be significant for very low accuracy levels.

A more demanding test case for the stability and the convergence of the iterated defect correction for the steady nonlinear Euler equations is the Ringleb’s flow [2]. It is one of the few continuous transonic flows of a blunt obstacle which can be solved analytically with the Hodograph method in a transformed $(V - \theta)$ plane, with V as the velocity magnitude and θ the angle of the velocity with respect to x -axis. More details on the Hodograph method and the analytical solution can be found in [2]. In our case the flow direction is upwards with the wall boundaries left and right. Their topology is derived from the analytical solution where the chosen boundaries of our test case represent two stream lines. The inflow and outflow boundaries are circles also given by the exact solution. The chosen geometry leans on th article [21].

In spite of being a transonic flow, it is smooth in the whole domain and since we can compute an exact solution at each grid point there is also a quantitative analysis possible. In addition the flow is supposed to be irrotational and isentropic. We performed the simulations on three successively adapted regular triangular grids imposing the exact solution on every boundary. Starting from the exact solution as

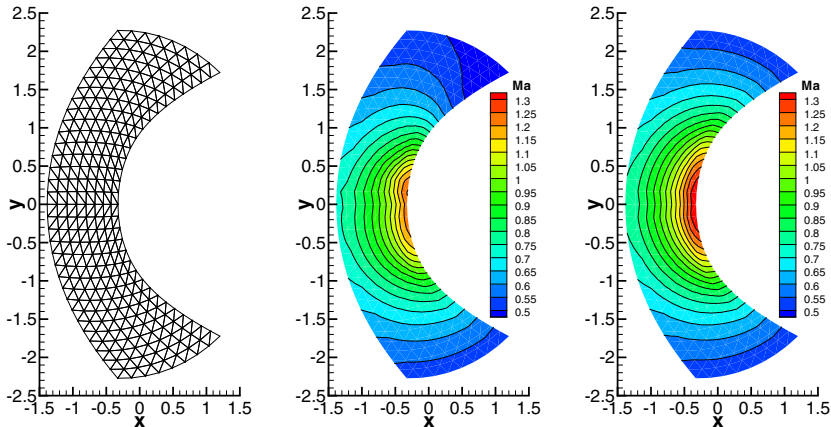


Fig. 5 One of the used triangular grids, steady solution of the first order basic method (middle) and defect correction solution with 4th order reconstruction (right)

the initial condition we use a first order scheme as the basic method for the iteration to a steady solution. In Fig. 5 the middle fine grid is depicted together with the steady solution of the basic method and the solution corrected with IDeC. We expect a complete symmetrical solution with a defined circular sonic line. The first order basic method clearly fails in these flow topologies. There are small instabilities, nevertheless the method is stable and converges perfectly, which is important for the method of iterated defect correction. For the defect correction we applied a 4th order accurate WENO reconstruction with the parameters $n_e = 2L$, $r = 6$, $\varepsilon = 10^{-14}$ and $\lambda_c = 10^5$.

With the IDeC we get a good solution which is near to the exact one in spite of the unsymmetrical solution of the basic method. With this setup we reach the theoretical convergence order of the reconstruction, proving the possibilities of the iterated defect correction method. If we compare the absolute error L^p -norms (Tab. 3) of both solutions this means a correction of the basic scheme from one up to four orders of magnitude for the finest grid.

Beside the nonlinear Euler equations we applied the method of iterated defect correction on the Navier-Stokes equations as well. Similar to the convergence studies shown before, we used the method of manufactured solution and solved

$$\mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{s}(\mathbf{x}) \quad (35)$$

with $\mathbf{F}(\mathbf{u}, \nabla \mathbf{u}) = \mathbf{F}^c(\mathbf{u}) - \mathbf{F}^d(\mathbf{u}, \nabla \mathbf{u})$, where \mathbf{F}^c and \mathbf{F}^d denote the convective respectively the diffusive flux. The defect correction formulation in (11) does not change in the case of solving the Navier-Stokes equation, but is just extended by the diffusive flux what results in

Table 3 Convergence rates for the first order basic method (up) and the corrected solution with 4th order reconstruction (down)

N	L^∞	L^1	L^2	$\hat{\mathcal{O}}_{L^\infty}$	$\hat{\mathcal{O}}_{L^1}$	$\hat{\mathcal{O}}_{L^2}$
Basic method $\mathcal{O}1$						
32	5.55E-02	5.89E-02	3.68E-02	-	-	-
64	3.56E-02	3.14E-02	2.01E-02	0.9	0.9	0.6
128	2.20E-02	1.64E-02	1.06E-02	0.9	0.9	0.7
Basic method $\mathcal{O}1 \rightarrow$ IDeC with $\mathcal{O}4$ reconstruction						
32	4.17E-03	9.55E-04	8.10E-04	-	-	-
64	3.23E-04	4.10E-05	4.04E-05	3.7	4.5	4.3
128	1.90E-05	2.07E-06	2.04E-06	4.1	4.3	4.3

$$R(\mathcal{Q}_i) = \int_{t^n}^{t^n+\Delta t} \int_{x_{i-1/2}}^{x_{i+1/2}} f^c(\mathcal{Q}_i)_x dxdt + \int_{t^n}^{t^n+\Delta t} \int_{x_{i-1/2}}^{x_{i+1/2}} f^d(\mathcal{Q}_i, (\mathcal{Q}_i)_x)_x dxdt \quad (36)$$

with \mathcal{Q}_i still acting as a placeholder for $u_i^{[k]}, u_i^{[k+1]}$ and $w_i^{[k]}$. Similar to the Euler equations the local defect is now computed for both fluxes, the convective and the diffusive flux. A high order formulation for the diffusive flux is therefore necessary. We have chosen the approximation suggested by Gassner et al. in [8]. It enables a one-step numerical method of high order accuracy in space and time using the same data as for the convection flux. Based on the idea of Godunov for advection problems not with constant initial data but with a linear initial distribution, it results in the so-called diffusive generalized Riemann problem (dGRP). Solving this Riemann problem yields two parts, the one containing the arithmetic mean value of the first derivative, whereas the second contains a physically motivated limiting term composed of the jump in the state of two adjacent cells. In 2D and 3D this leads to

$$\int_{\partial C_i} g(w_i^{[k]}, \nabla w_i^{[k]}) \mathbf{n} dS \approx \sum_{K \in \partial C_i} \sum_{j=1}^{n_{GP}} \omega_j^K g_{\mathbf{n}^K} \left(w_{i,j}^{[k]}, \left(w_{i,j}^{[k]} \right)_{\mathbf{n}^K} \right) S_K \quad (37)$$

with

$$\left(w_{i,j}^{[k]} \right)_{\mathbf{n}^K} = \frac{1}{2} \left(\frac{\partial}{\partial n} w_{i,j}^{[k],+} + \frac{\partial}{\partial n} w_{i,j}^{[k],-} \right) + \eta \left(w_{i,j}^{[k],+} - w_{i,j}^{[k],-} \right) \quad (38)$$

a numerical approximation for the diffusion flux, $\partial/\partial n$ denoting the derivative in normal direction. The characteristic length h is taken as twice the distance from the barycenter of the cell C_i to the barycenter of the edge K of the computed flux. The integration is again done by Gaussian quadrature with ω_j^K the weights on the edge K using a total number n_{GP} of integration points. The jump in the state of two neighboring cells is multiplied by the parameter η

$$\eta = \frac{1}{h\sqrt{\frac{1}{2}\pi}} \tag{39}$$

which can also be interpreted as a penalty term for the jump. With the chosen WENO reconstruction we obtain the derivative directly from the reconstructed polynomial distribution. However, the theoretical convergence order for this flux is limited to M , the degree of the reconstruction polynomials, in the case of a finite volume method. This is due to the fact that we use the first derivation of our polynomials losing hence one order of accuracy (see also [8]).

As an exact solution for the iterated defect correction applied on the Navier-Stokes equations simulated in two dimensions we took

$$\mathbf{u}_e = \begin{pmatrix} \sin(\pi x)\sin(\pi y) + 4 \\ \sin(\pi x)\sin(\pi y) + 4 \\ \sin(\pi x)\sin(\pi y) + 4 \\ (\sin(\pi x)\sin(\pi y) + 4)^2 \end{pmatrix} \tag{40}$$

with $\mathbf{u} = (\rho, \rho u, \rho v, \rho e)^T$ denoting the vector of the conservative state. By inserting (40) into (35) we can again compute the source term \mathbf{s} , which is only a function of the space \mathbf{x} . To test the method of iterated defect correction for rather viscous flows we chose a viscosity $\mu = 10^{-1}$, which results in a very low Reynolds number of $Re = 80$. As the temperatures are very low and do not take effect on the viscosity, we performed these computations with the assumption of a constant μ . The simulations were all carried out on a fully periodic domain $\Omega = [0; 2]_x[0; 2]$ with periodic boundaries on successively adapted regular triangular grids.

As we use the derivative of the polynomial distribution for the flux approximation we have to take a basis scheme with at least second order of accuracy for the defect correction. In Tab. 4 we show the convergence rates of the test case above computed with a second order basis method and corrected by a local defect using a 4th order reconstruction. Motivated by several assumptions found in the literature on the numerical error which is supposed to be dominated by the convection part we could think of dividing the local defect into an convective and a diffusive part. As both can be computed independent from another we performed the same simulation

Table 4 Convergence rates computed for the pressure for a second order basic method and a defect correction with 4th order reconstruction

h	L^∞	L^1	L^2	\mathcal{O}_{L^∞}	\mathcal{O}_{L^1}	\mathcal{O}_{L^2}
Basic method $\mathcal{O}2 \rightarrow$ IDeC with $\mathcal{O}4$ reconstruction						
0.200	2.05E-01	1.95E-01	1.23E-01	-	-	-
0.100	1.40E-02	1.36E-02	8.63E-03	3.9	3.8	3.8
0.050	8.87E-04	8.84E-04	5.54E-04	4.0	3.9	4.0
0.025	5.81E-05	5.49E-05	3.46E-05	3.9	4.0	4.0

Table 5 Convergence rates for the second order basic method (up) and the corrected solution with 4th order reconstruction (down) and a local defect computed only from the convection flux

h	L^∞	L^1	L^2	θ_{L^∞}	θ_{L^1}	θ_{L^2}
Basic method $\mathcal{O}2$						
0.200	6.69E-01	6.27E-01	3.99E-01	-	-	-
0.100	1.78E-01	9.66E-02	1.51E-01	1.9	2.0	2.0
0.050	4.66E-02	3.85E-02	2.44E-02	1.9	2.0	2.0
0.250	1.42E-02	1.04E-02	6.50E-03	1.9	1.9	1.9
Basic method $\mathcal{O}2 \rightarrow$ IDeC with $\mathcal{O}4$ reconstruction						
0.200	2.01E-01	1.94E-01	1.22E-01	-	-	-
0.100	2.15E-02	2.33E-02	1.41E-02	3.2	3.1	3.1
0.050	7.08E-03	8.00E-03	4.77E-03	1.6	1.5	1.6
0.250	2.91E-03	3.70E-03	2.18E-03	1.3	1.1	1.1

as before, with a local defect defined only in the convection flux setting the local defect of the diffusion to zero.

From the convergence rates in Tab. 5 one can see that for low Reynolds numbers, i.e. for flows dominated by the viscosity, it is indispensable to compute the local defect also for the diffusive fluxes to reach the optimal order of convergence. However, the absolute error norms of the corrected solution are lower than the ones of the basic method computed with second order accuracy. That means, by taking into account only the local defect of the convective flux we can not reach the full optimal convergence order but we obtain a slightly better solution than that computed with the basic method.

4.2 Application Test Cases

4.2.1 The RAE 2822 Profile

As a first application test case for the method of iterated defect correction we simulated the flow around the RAE 2822 profile in two dimensions. It is one of the official test cases of the project MUNA. We solve only the nonlinear Euler equations for this test case, so the grid we used is fully unstructured and contains about 18.000 elements with a relatively high discretized profile of 180 points per each half of the profile. The farfield is situated at 40 chord lengths and the profile is simulated as slip wall with the velocity normal to the wall set to zero. The flow was defined by the flow conditions of the so-called test case 9 with $Ma = 0.73$ and an incident angle $\alpha = 2.78^\circ$. This yields a transonic flow with a shock on the upper side of the profile (see Fig. 6 right). To compare the solution of the iterated defect correction we performed the simulation on the same grid with the numerical code of the DLR

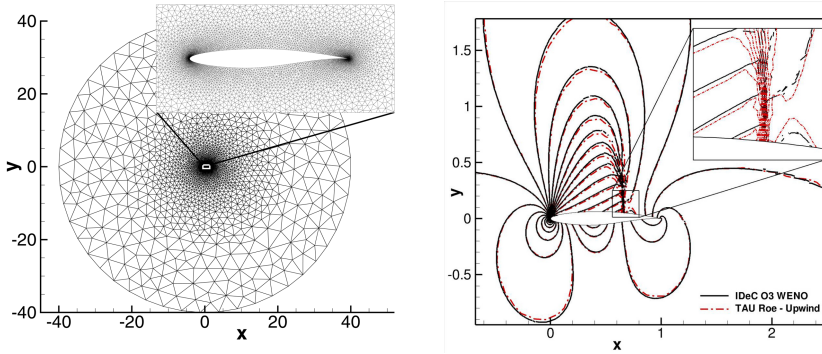


Fig. 6 Fully unstructured grid for the RAE 2822 profile (left) and Mach number distribution of the corrected solution and a direct solution with the TAU code using a second order TVD upwind scheme (right)

("Deutsches Zentrum für Luft- und Raumfahrt"), the TAU code, used as standard code for the project MUNA. The parameters for the TAU code were set to a second order TVD scheme with a least squares reconstruction and the Roe approximation for the convective flux.

In the case of iterated defect correction we used a first order basic method and corrected the steady solution by a local defect reconstructed with the WENO method described in section 3 using polynomials of degree two. The difference between the basic method and the corrected solution can be seen for the aerodynamic coefficients. In Fig. 7 we show the lift and drag coefficients over the number of iterations. Our iterative method is a rather suboptimal explicit method and so the number of iterations needed for a steady solution is quite high but does not influence the defect correction and is not of interest here. When the solution does not change any more we compute the local defect and solve afterwards the modified equations (5) to obtain a corrected solution which is denoted by the small arrows in Fig. 7. So each small arrow stands for a defect correction iteration.

We can see that the lift coefficient could already be corrected to the end solution after just one defect correction, whereas the drag coefficient needs some more defect correction iterations to converge. In the case of the lift coefficient a correction of about 11% was obtained and the drag coefficient could be corrected with even 28% of the first order solution, what corresponds to a total reduction of around 80 drag counts. The results of the TAU code serve here not only for validation but as a comparison as well, not having an exact solution for this test case.

In addition to the aerodynamic coefficients we can clearly see that by the iterated defect correction method with a 3rd order WENO reconstruction the shock is better resolved than it is with the second order least squares in the TAU code (see Fig. 6 right). Only if we once adapt the grid globally we are able to reach approximately the same shock resolution with the second order scheme.

4.2.2 Laminar Boundary Layer at High Reynolds Number

For the second application we solve the compressible Navier-Stokes equations at a low Mach number for a classical test case, the flow over a flat plate. We are simulating a laminar boundary layer but for a very high high Reynold number. Ludwig Prandtl and Blasius, one of his students, made pioneering achievements with their work on the boundary layer. This ended up in the solution of Prandtl’s boundary layer approximation equations by Blasius, reducing them to a nonlinear ordinary differential equation (ODE) of third order for the case of a laminar steady flow. This ODE can be solved nowadays numerically by a math algebra program with an arbitrary accuracy. We employed the software Matlab where we solved Blasius’ boundary layer equations (see e.g. [16]) with a four step Runge-Kutta scheme and a Newton-Raphson iteration method. This serves us as the reference solution for the iterated defect correction applied on the Navier-Stokes equations.

To point out the abilities of the WENO reconstruction in a reference space used in our work and the approach of the iterated defect correction we still used a fully unstructured triangular grid even in the boundary layer. For a finite volume scheme, this is a quite demanding task where surely some extra fine tuning is necessary to obtain satisfying results. One of them turned out to be the numerical flux approximation for which we took the HLLC flux as described in [20]. Our computational domain is $\Omega = [-0.5, 2] \times [0, 0.05]$ discretized by a total of 5250 triangular elements. In the interval $-0.5 < x < 0$ we use a slip wall boundary condition where the velocity normal to the wall is zero. At $x = 0$ we impose than a non-slip wall adiabatic boundary condition in the interval $0 < x < 2$. The free stream Mach number is $Ma_\infty = 0.3$, resulting from the free stream flow parallel to the wall with $u_\infty = 0.3$, $\rho_\infty = 1$ and $p_\infty = 1/\gamma$. As we use the equation of state for ideal gas the ratio of the specific heats is $\gamma = 1.4$, whereas the Prandtl number is $Pr = 1$.

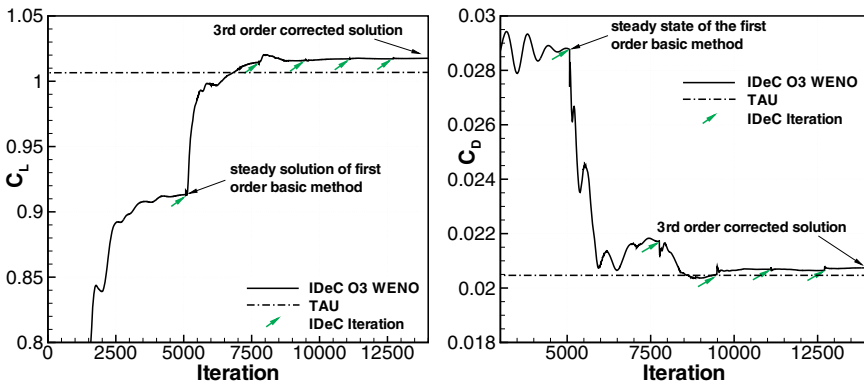


Fig. 7 Lift (left) and drag (right) coefficient computed with a first order basic method and a 3rd order defect reconstruction, compared with a second order TAU code solution on the same grid

For the chosen high Reynolds number of $Re = 10^6[1/m]$ we set our viscosity to $\mu = 3 \cdot 10^{-7}$ making again the assumption of a constant viscosity in the whole domain. To resolve the flow at high Reynolds numbers, which implies a very thin boundary layer of $\delta_{x=1} = 5 \cdot 10^{-3}$ in our case, a highly stretched grid in the boundary layer is necessary. At $x = 1$ we therefore have cells with an aspect ratio of 1 : 205. However the chosen spacing at the first cell of $y_1 = 4 \cdot 10^{-4}$ is still quite high compared to setups in the literature which are meant to be solved with a finite volume method 2nd order TVD method (see e.g. [9]). In addition we use only 8-9 cells to discretize the boundary layer at $x = 1$.

The initial condition is given by the free stream conditions and we take a homogeneous block profile with the free stream conditions at the inflow. It is important that mass can escape at the farfield, since due to the boundary layer growth we get a non-zero velocity outwards. At the outflow we can use simple extrapolation of the inner state values and impose only the free stream pressure. For the computation we used a second order basic scheme to obtain a steady solution which we reach after $t = 20$ seconds of simulation time. After each steady solution we apply the defect correction with a 4th order accurate WENO reconstruction again with the parameters $n_e = 2L$, $r = 6$, $\varepsilon = 10^{-14}$ and $\lambda_c = 10^5$. Left in Fig. 8 we see the distribution of the dimensionless x-velocity u/u_∞ in the whole domain for the basic method. In addition we compare the Blasius solution with the velocity profiles of both u and v components of the velocity at the position $x = 0.7$ (see Fig. 9) and with the skin friction coefficient over the entire plate (see Fig. 8 right). For all comparisons the corrected numerical solution, here shown after four defect correction iterations, is in good agreement with the Blasius reference solution. At this point, we have to mention, that using a scheme of higher order with WENO reconstruction is unstable and

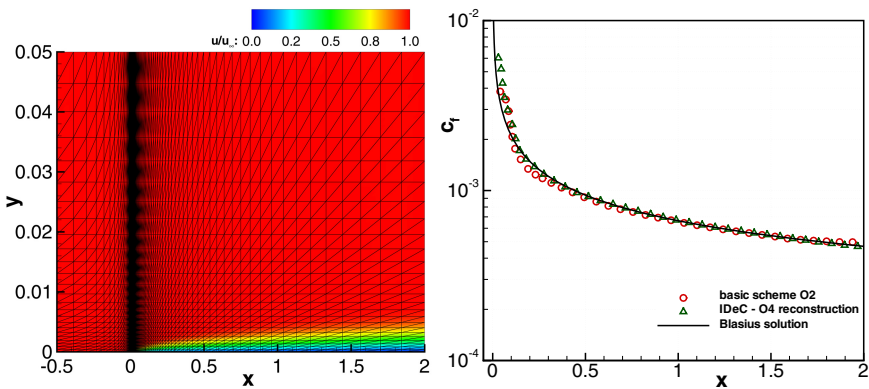


Fig. 8 Flat plate at high Reynolds number $Re = 10^6[1/m]$ with $\alpha = 0^\circ$ and $Ma = 0.3$ computed with a 2nd order basic method and corrected using a 4th order WENO reconstruction. Left we show the distribution of the dimensionless x-velocity u/u_∞ in the whole computational domain and right the skin friction coefficient is depicted over the plate length after four defect correction iterations.

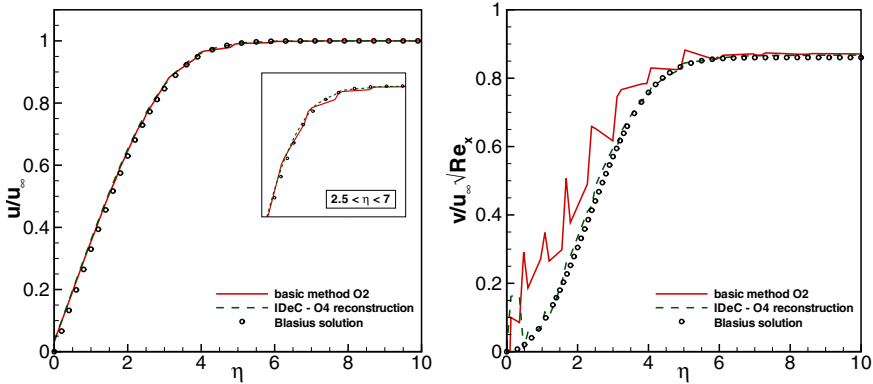


Fig. 9 Flat plate at high Reynolds number $Re = 10^6[1/m]$ with $\alpha = 0^\circ$ and $Ma = 0.3$ computed with a 2nd order basic method and corrected using a 4th order WENO reconstruction. Distribution of the dimensionless x-velocity u/u_∞ (left) and of the dimensionless y-velocity $v/u_\infty\sqrt{Re_x}$ (right).

no steady solution can be obtained. The method of iterated defect correction seems to be stable, as it is based only on a second order scheme, and is nevertheless able to correct the basic method with the high order accurate defect.

For the u component we can see in the zoom made in the section of the dimensionless variable $2.5 < \eta < 7$ with $\eta = y/\sqrt{\frac{\nu x}{u_\infty}}$ and the kinematic viscosity $\nu = \mu/\rho$, that we get a smoother distribution by applying the defect correction. In the case of the dimensionless v component of the velocity $v/u_\infty\sqrt{Re_x}$ with $Re_x = \sqrt{\frac{u_\infty x}{\nu}}$ the improvement of the solution is considerable. The completely wrong distribution of the basic method could be corrected to fit quite well with the reference Blasius solution. Near the wall we nevertheless reach the limits of the reconstruction, which has shown so far that the reconstruction in the reference space can cope even with these highly stretched elements. The corrected skin friction coefficient (Fig. 8 right) shows also better agreement with the reference solution especially at the beginning and the end of the plate. The remaining difference is due to the stagnation point at $x = 0$ where high gradients occur, causing oscillations. As can be read in literature the prediction of skin friction coefficients still remain a difficult issue in the numerical simulation. Similar to [9] we compare therefore the friction coefficients at the end of the plate at $x = 2$. The analytical solution can again be computed by solving the Blasius equations and we get

$$c_f = \frac{0.664}{\sqrt{Re_x}} \quad \text{with} \quad Re_x = \frac{\rho u_\infty x}{\mu}. \tag{41}$$

With $c_f(x = 2) = 4.983 \cdot 10^{-3}$ the error of the skin friction of the basic method is of 6.1% and with the iterated defect correction approach it could be corrected to $c_f(x = 2) = 4.653 \cdot 10^{-3}$ resulting in a remaining error of 0.9% for this case.

4.3 Conclusion

In this work we applied the method of iterated defect correction to a finite volume scheme and solved both, the Euler and the Navier-Stokes equations. A steady solution of lower accuracy, mostly of first or second order, is the starting point for the method of iterated defect correction. The next step consists of a WENO reconstruction which is used to evaluate the local defect of the steady solution. If we modify our equations by putting the local defect on the right hand side as a negative source term, the low order solution can be iteratively shifted to the accuracy of the reconstruction. The main advantage of this approach is that the high order scheme has not to be solved - the high order scheme is only used to calculate an estimation of the local discretization error. Hence, this approach can be used to increase the accuracy of an existing code in a straightforward way. It seems that the high order approximation also inherits some additional stability from the low order solver. Our results show that the iterated defect correction in combination with the WENO reconstruction in [4] for unstructured meshes works very well. We did not succeed to define robust boundary conditions in any case. This seems to be even more subtle than the definition of high order boundary conditions in general.

We have shown numerical convergence results up to sixth order of accuracy, applying the method of iterated defect correction starting with a first order steady solution. By modifying the original approach, a relevant speed up could additionally be achieved for equations with source terms depending on the solution itself. The scheme remained stable even for the more challenging test case of the transient Ringleb's flow. A fourth order accurate solution could be achieved here from a first order numerical scheme. Convergence studies using the manufactured solutions method have shown that in the case of the Navier-Stokes equations, where a local defect can be computed separately for the convective and the diffusive fluxes, it is crucial for flows with low Reynolds numbers to evaluate the local defect for all fluxes. Neglecting the defect in the convective fluxes still gave better absolute error norms compared to the low order solution, but the expected high order of the reconstruction was not reached for our test cases.

A RAE 2822 profile have been computed as an application test case, solving the Euler equations with a first order basic scheme applying a third order accurate reconstruction to determine the local defect. Compared to the first order solution the aerodynamic coefficients like lift and drag could be corrected by 11%, respectively 28%. As a second application test case a laminar boundary layer at a high Reynolds number of $Re = 10^6$ was numerically solved using the iterated defect correction. Better results compared with the second order starting solution could be achieved applying the defect correction combined with a fourth order reconstruction. For example a reduction of the skin friction error at the end of the plate from 6.1% with the basic scheme to 0.9% was reached in this case.

References

- [1] Abgrall, R.: On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics* 144, 45–58 (1994)
- [2] Chiocchia, G.: Exact solutions to transonic and supersonic flows. AGARD Advisory Report AR-211, pp. 1–14 (1985)
- [3] Cockburn, B., Karniadakis, G., Shu, C.: *Discontinuous Galerkin Methods. Lecture Notes in Computational Science and Engineering*. Springer (2000)
- [4] Dumbser, M., Käser, M.: Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics* 221(2), 693–723 (2007)
- [5] Frank, R.: The method of iterated defect-correction and its application to two-point boundary value problems (part 1). *Numer. Math.* 25, 409–419 (1976)
- [6] Frank, R.: The method of iterated defect-correction and its application to two-point boundary value problems (part 2). *Numer. Math.* 27, 407–420 (1977)
- [7] Frank, R., Ueberhuber, C.W.: Iterated defect correction for differential equations part i: theoretical results. *Computing* 20, 207–228 (1978)
- [8] Gassner, G., Lörcher, F., Munz, C.D.: A contribution to the construction of diffusion fluxes for finite volume and discontinuous galerkin schemes. *Journal of Computational Physics* 224, 1049–1063 (2007)
- [9] Hirsch, C.: *Numerical Computation of Internal and External Flows*, vol. 1. Wiley (1989)
- [10] Hu, C., Shu, C.: Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics* 150, 97–127 (1999)
- [11] Jiang, G., Shu, C.: Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 202–228 (1996)
- [12] Käser, M., Iske, A.: ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics* 205, 486–508 (2005)
- [13] Liu, X., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. *Journal of Computational Physics* 115, 200–212 (1994)
- [14] Pereyra, V.: Iterated deferred corrections for nonlinear boundary value problems. *Numer. Math.* 11, 111–125 (1968)
- [15] Pereyra, V.L.: On improving an approximate solution of a functional equation by deferred corrections. *Numer. Math.* 8, 376–391 (1966)
- [16] Schlichting, H.: *Boundary Layer Theory*, 7th edn. Springer (1979)
- [17] Shu, C.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. NASA/CR-97-206253 ICASE Report No.97-65 (1997)
- [18] Stetter, H.: Economical global error estimation. In: Willoughby, R.A. (ed.) *Proceedings of the International Symposium on Stiff Differential Systems*. Plenum Press, New York (1974); Wildbad, Germany (1973)
- [19] Stetter, H.J.: The defect correction principle and discretization methods. *Numer. Math.* 29, 425–443 (1978)
- [20] Toro, E.: *Riemann solvers and numerical methods for fluid dynamics*. Springer, Heidelberg (1997)
- [21] Wang, Z., Liu, Y.: Extension of the spectral volume method to high-order boundary representation. *Journal of Computational Physics* 211, 154–178 (2006)

- [22] Zadunaisky, P.: A method for the estimation of errors propagated in the numerical solution of a system of ordinary differential equations. In: Proceedings of the International Astronomical Union, Symposium, vol. 25. Academic Press, New York (1966)
- [23] Zadunaisky, P.E.: On the estimation of errors propagated in the numerical integration of ordinary differential equations. *Numer. Math.* 27, 21–39 (1976)

Part IV
Geometry and Deformation

Uncertainties of Numerical Structural Models in the Frame of Aeroelasticity

P. Reich, A. Reim, M. Haupt, and P. Horst

Abstract. Today, numerical methods for structural and aerodynamic problems are reaching highly versatile and reliable levels. Therefore, the coupling of both domains can be solved at a high standard. On the other side, the accuracy of aeroelastic analyses depends on the level of precision with which the stiffness properties and, thus the structural behavior of an aircraft wing structure in means of deformation can be predicted. The presence of uncertainties within the structural model which is integrated in the coupled analysis can affect the fidelity of the structural response and, thus, influence the results of the numeric aerodynamic simulation as well. Investigations carried out by the Institute of Aircraft Design and Lightweight Structures (IFL) in the frame of the MUNA-project were focused on two types of uncertainties affecting the accuracy of the static aeroelastic analysis: stochastic uncertainties and uncertainties due to modeling simplifications. Stochastic uncertainties are caused by the deviation of actual structural parameters in realized aircraft wings, like Young's modulus or wall thicknesses from the original ideal design. This deviations affect the stiffness of the real structure and, thus the structural and aerodynamic response. A method to estimate the sensitivity of the wing structure to random input parameters is presented in the second part. The second class of uncertainties arises from approximations connected to the idealization of the physical and geometric properties of the real structure used in finite element (FE) structural models. In the first part of this work, an overview of modeling effects is given which affect the stiffness properties of the FE structural models and in turn influence the results of static aeroelastic analysis. The coupled analysis is carried out with a high-order panel method for the aerodynamic domain and a parametric finite element structural model, which allows a wide variation of material and geometric properties of wing box structure. This structural model as well as the aerodynamic method and the coupling routines are presented in the following section.

P. Reich, A. Reim, M. Haupt, and P. Horst
IFL, Hermann-Blenk-Str. 35, D-38108 Braunschweig
e-mail: paul.reich@tu-bs.de

1 Finite Element Models and Analysis Methods Used for Uncertainty Quantification

1.1 Parametric Finite Element Model

A code already developed at the Institute of Aircraft Design and Lightweight Structures (IFL) is enhanced to generate a finite element model of the structure. It is based on the parametric description of airplane wing geometry and a layout of the load-bearing structure [1], [2]. The code is written in Patran Command Language (PCL) which allows an automated generation of finite element wing models by the preprocessor MSC Patran[®].

A HIRENASD wind tunnel model [3] scaled up to 58 m of span is employed as a test structure for investigations carried out in the context of the MUNA project. The wing box structural layout and the arrangement of engines are taken on from the predecesing project [4] and resemble the wing of an AIRBUS A340 aircraft (see Fig. 1).

The geometric data are imported from an ASCII input file and are employed to generate a finite element shell model of the wing. A transonic transport aircraft design is used with the corresponding weights given in table 1 to evaluate the target lift for the calculation of aerodynamic and static inertial loads.

Due to a high number of required aeroelastic calculations, especially for the stochastic analysis presented in the second part of the work, the high order panel method HISSS is used instead of an Euler or RANS code to calculate the discrete aerodynamic nodal loads. The lack of accuracy when calculating a load distribution on the wing surface at higher Mach numbers had to be accepted so that the numeric costs could be kept reasonable. The finite-element solver NASTRAN was used to calculate the nodal displacements of the structural model.

The in-house code coupling library ifls [5] was employed to perform the fluid-structure interaction. The code handles the load and displacement transfer between non-conform grids by using a three-field approach in combination with Lagrange multipliers. The structure of the coupling routine allows the interaction between different commercial numerical solvers. The converged angle of attack α_{EqsI} of static aeroelastic equilibrium was estimated by ifls iteratively for given lift and flow conditions by variation of an overall (geometric) angle of attack α_g of the wing.

Table 1 Weights for the transonic transport aircraft design used in this study

Gross weight	m_{TOW}	to	256
Fuselage and empennage unit structure + payload	$m_{RF} + m_N$	to	95
Wing structure	m_W	to	35
Total fuel mass	m_F	to	106
Propulsion group	m_{PG}	to	20

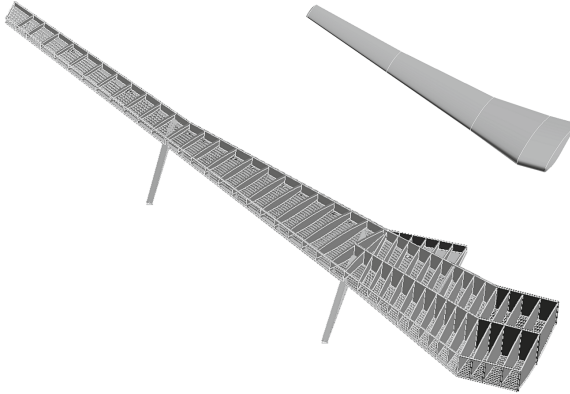


Fig. 1 HIRENASD wing geometry and structural layout

Nodal loads calculated on the aerodynamic surface were transferred to the nodes of the structural grid by means of conservative interpolation in the area of the wing box. In the region of the flap and slat structure the aerodynamic nodal loads were applied to additionally created auxiliary structural nodes and tied to the wing box by multi point constraints of RBE3 type.

The static inertial loads including fuel weight, engine loads and the weight of the flap structure had also to be taken into account to represent realistic loading conditions. The flap and slat structure were idealized as point masses and connected to the spar structure by multi point constraints in the same way as the aerodynamic forces. The masses of the high lift devices were also required for this idealization and were estimated by handbook methods [6]. Tank loads were also modeled with point masses and RBE3s. The tank masses were evaluated for each wing bay by calculation of the volume taken by the fuel for a given degree of refueling.

The static inertial loads including fuel weight, engine loads and the weight of the flap structure had also to be taken into account to generate realistic load cases. The flap and slat structure were idealized as point masses and tied to the spar structure by multi point constraints in the same way as the aerodynamic forces. The masses of the high lift devices needed for this simplified approach were estimated by handbook methods [6]. Tank loads were also modeled with point masses and RBE3s. The tank mass was estimated for each wing bay by calculation of the volume taken by the fuel for a given degree of refuelling.

The wing box structure was sized with respect to strength criteria and constraints of buckling stability. Two load cases were selected for the sizing process: a 2,5g maneuver and the landing impact (see table 1). The strength sizing was carried out by a fully stressed design using stress distribution computed for limit loads and a yield-stress criterion. The design against buckling failure was performed by handbook methods [7] using maximal allowable stresses for the compression panels

Table 2 Weights for the transonic transport aircraft design used in this study

2.5g maneuver			
Altitude	H	km	11
Mach number	Ma	-	0.82
Gross weight	m_{TOW}	to	256
landing impact			
Altitude	H	km	0
Mach number	Ma	-	0.2
Gross weight	m_{TOW}	to	182
Cruise flight (1g)			
Altitude	H	km	11
Mach number	Ma	-	0.82
Gross weight	m_{TOW}	to	256

as well as optimum design curves and semi-empirical formulas for estimation of stiffener spacing and cross-section geometry.

Due to constraints defining the highest permitted elastic deflection of the wing tip given in [4], the wing box was also sized under consideration of stiffness. For this additional sizing procedure the contribution of structural members to the wing deflection was calculated following the pattern of the modified fully utilized design method (MFUD) proposed by Patnaik et al [8]. For the constrained degree of freedom (in this case it is the bending displacement) the sensitivity factors can be calculated for each component of the structure. These factors are defined as dw/dm where dw is a partial change of displacement and dm is a change of structural mass. The change of bending deformation and structural mass are evaluated by attaching additional material (by increasing wing thickness or stiffener cross-section) to each structural member and recalculating the displacement w of the modified structure subjected to a reference load case. These sensitivity factors are used within the MFUD-procedure to weigh the increase of wall thickness of the structural members until the displacement constraint is achieved. This method permits to attach an additional structural mass only in those areas of the wing box whose stiffness influences the given deformation at most. The weight of the structure sized with this approach was estimated to be very close to those obtained by a time-consuming optimization procedure [8].

1.2 Quantification of Uncertainty

For the quantification of uncertainty affecting the static aeroelastic response different structural and aerodynamic output parameters are considered. Evaluation criteria commonly used for characterization of the aeroelastic response are the lift and drag coefficients for a given angle of attack, natural frequencies, or flutter speed of

the investigated aircraft. In this study only static aeroelasticity is treated, therefore the study is concerned with the aerodynamic performance of an aircraft wing under cruise flight conditions (see table 2). For this purpose the converged angle of attack α_{EqSt} is evaluated iteratively for a given lift. A coupled fluid structure analysis is performed using derivatives of the wing structural model presented in section 1.1 which is affected by different types of uncertainties. For each derivate relative deviation $\Delta\alpha_{EqSt}/\alpha_{EqSt}$ compared to the result obtained for a reference structure (without modifications) is calculated.

The global values for relative difference to the converged angle of attack $\Delta\alpha_{EqSt}/\alpha_{EqSt}$ presented in this work are influenced not only by the change of structural parameters, but are dominated by the aerodynamic properties of the wing as well as by the given flow conditions and aerodynamic method used within the static aeroelastic analysis. For this reason, the results presented within this work should be considered as sample values to demonstrate the degree of deviation within aerodynamic output parameters for a special test case.

To examine the change within the wing box stiffness the structural response (without aeroelastic coupling) is calculated for different derivatives of the FE test model subjected to a reference load case. The reference load case is represented by a pressure distribution and inertia loads obtained for a reference structure under cruise flight conditions. Local values of bending angle $w'(y)$ and twist $\Theta(y)$ are computed along the structural wing span. These "beam-like" deformations are extracted from the nodal solution of the 3D finite-element model by means of the method presented by Malcolm and Laird [9]. The procedure employs a least squares fitting to extract three translational and three rotational section deformations from the nodal displacements in x -, y - and z -direction for each wing section. This process is applied to a series of sections along the wing span to calculate the bending and torsion. For the local values of bending and torsional angle the deviations $\Delta w'(y)$ and $\Delta\Theta(y)$ are calculated relative to the deformations obtained for the reference structural model. The local deviations are related to the maximum reference values of the corresponding deformation, $w'(y)_{max}$ and $\Theta(y)_{max}$ respectively. This approach ensures that no singularities can occur due to very small local values within the torsion deformation.

To estimate the effect of stiffness variation on the wing aerodynamics, a well-known concept for the elastic angle of attack α^{el} is used. This kinematical term describes the local change of the geometric angle of attack α^g in flight direction due to elastic deformation of the wing. It affects the load distribution caused by the flexible structure of lifting surface and thus the overall lift coefficient. Deviations in torsion and bending stiffness of the wing box cause a change of the lift distribution over the wing span compared to the reference structure. Under conditions of steady cruise flight this lift change must be corrected by adapting the angle of attack α^g of the aircraft iteratively until target lift will be achieved and $\alpha_{EqSt} = \alpha^g$.

For swept wings the local angle $\alpha^{el}(y)$ depends on the torsion deformation $\Theta(y)$ as well as on the bending angle $w'(y)$:

$$\alpha^{el} = \Theta \cos \varphi - w' \sin \varphi \quad (1)$$

From the kinematical interrelationship in equation (1) follows that for a wing with positive angle of sweep back φ the torsion and bending contributions of the elastic angle of attack are directed mutually. For this reason, the change of bending angle due to reduction of bending and shear stiffness of the wing box structure can be compensated by the change of torsion deformation caused by the reduced torsion stiffness to a certain degree. For common transonic transport aircraft wing structures the angle α^{el} is dominated by the bending deformation and for this reason is negative.

To estimate the effect of the variation of torsion and bending distortions on the deviation of the elastic angle the propagation of uncertainty is applied on equation (1). For a local relative deviation $\Delta\alpha^{el}(y)/\alpha_{max}^{el}$ in elastic angle of attack a mathematical correlation (2) is the following:

$$\frac{\Delta\alpha^{el}(y)}{\alpha_{max}^{el}} = \frac{\Delta\Theta(y)}{\Theta_{max}} \left(\frac{\Theta_{max}}{\alpha_{max}^{el}} \right) \cos\varphi(y) - \frac{\Delta w'(y)}{w'_{max}} \left(\frac{w'_{max}}{\alpha_{max}^{el}} \right) \sin\varphi(y) \quad (2)$$

The local values $\Delta\Theta(y)/\Theta_{max}$ and $\Delta w'(y)/w'_{max}$ are relative deviations of torsion and bending distortions due to the local change of wing box stiffness caused by different degrees of modeling simplification. These values are structural parameters depending on the stiffness properties of the structure. The terms $\Theta_{max}/\alpha_{max}^{el} \cos\varphi$ and $w'_{max}/\alpha_{max}^{el} \sin\varphi$ in equation (2) are ratios of the local torsional and bending angles relative to the maximum value of elastic angle of attack. These values depend on the local sweep back angle $\varphi(y)$ of the wing box reference axis, the load distribution in chord and span-wise directions (ratio of the distributed moment relative to the distributed load) as well as on the ratio of the torsion stiffness GJ relative to the bending stiffness EI .

The local deviations of the elastic angle of attack $\Delta\alpha^{el}(y)/\alpha_{max}^{el}$ are related to the maximum value obtained for the reference FE model in the same manner like deviations of structural deformations. Since the effect of the deviation of this parameter on the geometric angle of attack α^s and thus on the local lift distribution is depending on the magnitude of $\alpha^{el}(y)$ this approach seems to be more suitable for the objective of the present study than relating this term to the reference local values as commonly done. The latter method would overestimate the influence of the deviation $\Delta\alpha^{el}(y)$ considering local variations of the elastic angle of attack near the root as well, which have no appreciable effect on the load distribution due to the very small values of $\alpha^{el}(y)$ within this area.

The distribution of the local deviations $\Delta\Theta(y)/\Theta_{max}$, $\Delta w'(y)/w'_{max}$ and $\Delta\alpha^{el}(y)/\alpha_{max}^{el}$ varies along the wing structural axis, depending on the stiffness and load distribution of the present wing structure and aerodynamic design. To obtain global deviation parameters the mean values of these local variations are calculated in sections using a relation defined exemplarily in equation (3) for the bending angle:

$$[\Delta w'] = \frac{1}{s} \int_0^s \frac{\Delta w'(y)}{w'_{max}} dy \quad (3)$$

where s is the structural span of the wing box. Mean values of twist $[\Delta\Theta]$ and elastic angle of attack $[\Delta\alpha^{el}]$ are obtained in the same way. To assess the contribution of the variations of torsional and bending angles to the deviation of elastic angle of attack, the local values $\Delta w'(y)/w'_{max}$ and $\Delta\Theta(y)/\Theta_{max}$, multiplied with the parameters' terms $w'_{max}/\alpha_{max}^{el} \sin\varphi$ and $\Theta_{max}/\alpha_{max}^{el} \cos\varphi$ from equation (2), are integrated by means of eq. (3). These "transformed" values $[\Delta w']_{tr}$ and $[\Delta\Theta]_{tr}$ are also used within the present work to estimate the effect of variation within structural stiffness properties caused by modeling or stochastic uncertainties on the load distribution. The difference between the term $[\Delta\alpha^{el}] = [\Delta\Theta]_{tr} - [\Delta w']_{tr}$ calculated from the global bending and torsion deformations and the mean value resulting from integration of the local values $\Delta\alpha^{el}(y)/\alpha_{max}^{el}$ directly obtained from the structural response is between 0.01% and 0.03%.

Because nonlinear behavior of aeroelastic problems is highly depending on the local flow conditions as well as on local stiffness characteristics of the wing structure, the change in equilibrium state angle of attack $\Delta\alpha_{EqSt}/\alpha_{EqSt}$ cannot be predicted using a mean value $[\Delta\alpha^{el}]$ for a complex structure in a direct way. Nevertheless, as will be shown within the following sections, the parameter $[\Delta\alpha^{el}]$ is a suitable indicator to estimate the deviation tendency of wing aerodynamics due to the variation within the stiffness properties of the wing.

2 Part I: Model Uncertainties

2.1 Introduction

To obtain a high level of accuracy for a structural model one possible approach is to reproduce the real structure with a high level of geometric detail. This approach implies two general drawbacks: it is connected with high modeling effort on the one hand and requires fine discretization of the wing box geometry on the other hand (see Fig. 4 on the left hand side), resulting in high model size and numerical costs. To demonstrate the dimension of complexity connected with a detailed model the reference structure described in section 1.1 is considered. The FE model has 740 design variables and is realized by 24900 shell and 10700 bar elements having in total 260000 degrees of freedom.

For coupled aeroelastic analysis, requiring a high number of iterations the minimization of the finite element model size could be of high priority. As first approach to reduce the number of degrees of freedom, reduction techniques are used to condense a 3D wing box structural model into a 1D-beam stick model. In the other case, if a parametric FE wing model with variable geometry should be optimized, the amount of design variables associated with high level of modeling detail is undesirable. To reduce the number of design variables a simplified structural model is preferred which is composed only of main components of the wing structure, like top and bottom covers, ribs and spars, including spar caps. Within such

simplified models stringer stiffeners are commonly idealized by an additional layer with orthotropic material properties. The next sections deal with the effect of the simplifications within the 3D FE models.

2.2 *Design of the Wing Structure*

The structure of a common transport aircraft wing is composed of the following components (see fig. 2):

- Spars with spar webs carrying shear load and spar caps which resist tension or compression normal loads. Generally, the wing box is composed of a front and rear spar. The rear spar is of special importance for the mounting of movables as well as for systems integration. As shown in fig. 2, additive components, like a mid-spar, or a false rear spar can be integrated in the wing structure as well.
- Top and bottom covers which have a contouring as well as a load carrying function. These components carry both, normal and shear stresses. Along with main spars, the wing skin forms the box structure of the wing. The skin parts are stiffened by stringers to prevent buckling failure
- Ribs which can be oriented perpendicular to the wing box axis or parallel to the aircraft symmetry plane. Ribs are used to keep the aerodynamic shape of the wing cross-sections under aerodynamic load and for insertion of concentrated loads in the wing box structure caused by engine mountings or landing gear.

In figure 2 different levels of modeling details for each structural component are depicted. The stiffening components, like stringers, spar- or rib caps can be realized by beam elements with defined cross-sections, by rod elements neglecting the bending stiffness of the stiffener or can be "smeared" over the area of the correspondent thin-walled structural component. The smeared stiffening component in turn can be realized as an additional orthotropic material layer, which is reasonable when modeling the skin-stringer panels or taken into account in the wall thickness of the thin-walled components what is commonly done by ribs and spars.

In the current study the structural model is realized by shell and bar elements. The inclusion of element offsets was used as a reference with highest grade of modeling detail limited to the main stiffening components within the present work.

The different levels of detail, shown in figure 2 are employed within the test wing model. The idealized FE models are derived from the reference geometry by replacing the built-up structure, realized with shell and bar elements, by a.m. simplified structural design. To estimate the deviations in the deformation behavior caused by modeling simplifications the structural response of both the reference and simplified wing box models is compared for a reference loading. To ensure the comparability of these results the volume of the wing structure was kept constant for all derivations with varying grade of the detail.

Uncertainties caused by different levels of approximation are discussed in the following section. The intent of this overview is not to enable the general prediction

of the modeling error of the structural and thus of the aeroelastic response resulting from distinct simplification. Due to the individual design and stiffness properties of wing structures realized in different aircraft types the contribution of structural components to the bending, shear, and torsional stiffness as well as to the warping characteristics varies depending on a given structural design. Instead of that, a series of calculations is performed to estimate the dimension of the deviations resulting from different levels of simplification using the parametric finite element model. The effect of uncertainty on the stiffness properties of the wing structure is considered only in the frame of static aeroelastic analysis under assumption of linear elastic structural behavior. Non-linear effects as well as dynamic properties or effects caused by the usage of non-isotropic materials are not in the objective of the present work.

2.3 Uncertainties due to Modeling Simplifications

A series of comparing analyses is carried out to estimate the influence of geometrical details on the accuracy of wing structural response. Global deviation parameters presented in section 1.2 are used to evaluate at first the change in structural stiffness components due to modeling approximations and secondly the influence of these alterations on the static deformation of the wing in flight direction considering the elastic angle of attack α^{el} . For selected cases a static aeroelastic analysis is carried out to calculate the deviation of equilibrium state angle of attack and, therefore, to estimate the impact of altered structural stiffness on the aeroelastic response. To

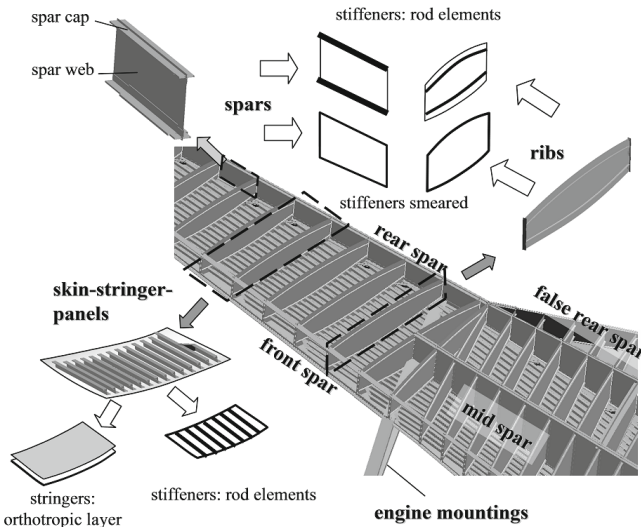


Fig. 2 Components of the wing box structure

distinguish the deviations of structural stiffness five integral parameters are listed in tables 3 and 4 for each modeling effect. These parameters are the integral deviation of bending and torsional angles $[\Delta w']$ and $[\Delta \Theta]$ relative to the elastic axis of the wing, the "transformed" values $[\Delta w']_{tr}$ and $[\Delta \Theta]_{tr}$ of both angles in flight direction as well as the global deviation of elastic angle of attack, $[\Delta \alpha^{el}]$.

2.3.1 General Simplifications

Effect of the Element Offset

To reproduce the outer mold surface of the real wing structure the shell elements forming the wing skin must have an offset relative to the nodes of the discretized geometry. By neglecting the element offset the distance of skin panels relative to the neutral axis of the wing box will be overestimated. This effect will increase the moment of inertia of the wing box cross-section following the parallel axis theorem, which in turn results in higher bending stiffness of the wing structure compared to the exact solution. The torsional stiffness will also be affected by the increasing distance between the mid-lines of the top and bottom covers in accordance with the Bredt-Batho formulation. The same effect on the bending stiffness appears by ignoring the offset distance of beam or rod elements representing the stiffening structural members. The latter case will be considered separately for each stiffening component.

To estimate the impact of the modeling simplifications on the deformation behavior of the wing box the structural response is calculated for the idealized and the reference structural models. The integral values of the deviation in bending angle, torsion and resulting elastic angle of attack compared to the reference structure are given in table 3. For the FE wing model without element offset within the top and bottom covers the bending stiffness increases accordingly to the a.m. effects resulting in an approximately 4% smaller bending angle which in turn reduces the local angle of attack (cp. section 1.2). Torsion deformation is also reduced by 1.7% due to the higher torsional stiffness having an opposite effect. The change in both degrees of freedom results in 4.1% smaller elastic angle of attack due to the dominant influence of the bending stiffness (compare the values $[\Delta w']_{tr}$ and $[\Delta \Theta]_{tr}$ in table 3). The sign of the transformed deviation parameter $[\Delta \Theta]_{tr}$ changes due to the relation to the maximum value of elastic angle of attack α_{max}^{el} .

The converged angle of attack calculated for the more simplified structure shows a 0.9% smaller value compared to the reference model (see fig. 5). This result corresponds with the trend predicted by the negative change of the elastic angle of attack $[\Delta \alpha^{el}]$ given in table 3. Smaller (negative) values of $\alpha^{el}(y)$ along the span have a reduced effect on the load distribution compared to the reference structure and, therefore, the target lift can be achieved under smaller angle of attack.

Table 3 Deviations of bending angle, torsional angle and elastic angle of attack for different states of modeling simplification

	shell elems. without offset		simplified BCs	
	rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.
$[\Delta w']/[\Delta w']_{tr}/\%$	-3.955	-4.327	-5.446	-5.782
$[\Delta \Theta]/[\Delta \Theta]_{tr}/\%$	-1.699	0.228	-29.534	3.974
$[\Delta \alpha^{el}]/\%$		-4.101		-1.806

Simplified Boundary Conditions

At the root, the wing is mounted to the wing center box and to the main frames of the fuselage. Despite of high local wing box stiffness, a minimal translational displacement of the wing skin in span wise direction is possible in the root area. If this infinitesimal displacement is constrained by restriction of all translational and rotational degrees of freedom along a root rib curve (see fig. 3 on the right hand side), a reduction of bending and torsion deformations due to the overestimation of wing root rigidity can appear. This kind of idealization is used when the structure of a half wing is realized without the center box. To assure realistic boundary conditions, see figure 3 on the left hand side, the displacement of upper and lower edges of the root rib, should be constrained only in direction normal to the skin surface (z-direction). The nodal displacements in spanwise direction (along the y-axis) as well as nodal rotations have to be constrained only at the symmetry plane of wing center box.

The effect of higher wing root rigidity has a local character influencing the bending and torsional deformations in the form of additional (negative) rigid body motions, resulting in the integral deviation of 5.4% within the bending and approx.

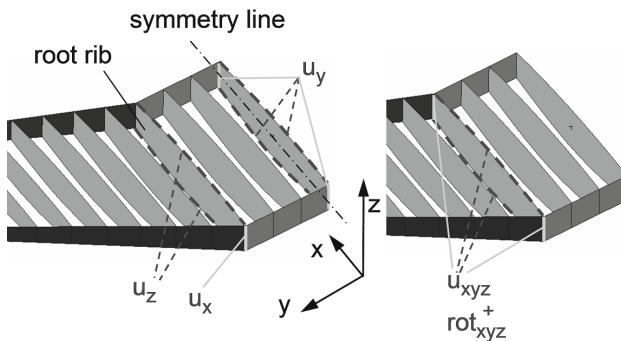


Fig. 3 Realistic and simplified boundary conditions

30% within the torsional angle. Due to the mutually directed influencing tendencies of these deformations, the resulting change within the elastic angle of attack is only 1.8% (see table 3). From the transformed values $[\Delta w']_{tr}$ and $[\Delta \Theta]_{tr}$ in table 3 it can be seen that the rather high contribution of bending deviation in flight direction is compensated by the much higher change of the torsional angle.

The results of the coupled aeroelastic analysis confirms with the tendency of the deviation of elastic angle of attack $\alpha^{el}(y)$ obtained by the structural response. This moderate tendency of the change of is reflected in the deviation of the equilibrium state angle of attack $\Delta \alpha_{EqSt} / \alpha_{EqSt}$ being only 0.54% (see fig. 5).

2.3.2 Idealization of Stiffened Structural Components

In the following sections, the effects of the different degrees of detail of modeling are discussed. Several types of stiffener idealization are considered and the effect of the simplifications on the structural behavior and aerodynamic properties of the wing is evaluated by calculating structural and static aeroelastic response. The deviations of structural response computed for each case are summarized in a test matrix (see fig. 4). In the test matrix, different degrees of modeling detail are considered for stringers, spar caps and rib caps. The levels of modeling detail are represented by realizing the structural member by beam elements or rod elements, or by homogenizing the stiffeners as isotropic or orthotropic layer. The effect of element offset is also considered for beam and rod elements as well as for the orthotropic material layer. For the main idealizations, the deviation of converged angle of attack from the reference case is plotted in figure 6.

		stringers		spar caps		rib caps	
		rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.
bar elems. w-out offset	$[\Delta w'] / [\Delta w']_{tr} \%$	-5.787	-6.442	-0.548	-0.601	-0.0131	-0.0149
	$[\Delta \Theta] / [\Delta \Theta]_{tr} \%$	-0.224	0.032	-0.455	0.061	0.0172	-0.0029
	$[\Delta a^{el}] \%$		-6.409		-0.540		-0.0178
rod elems. + offset	$[\Delta w'] / [\Delta w']_{tr} \%$	0.843	0.940	-0.041	-0.044	0.0375	0.0408
	$[\Delta \Theta] / [\Delta \Theta]_{tr} \%$	0.476	-0.073	-0.402	0.055	-0.0896	0.0140
	$[\Delta a^{el}] \%$		0.866		0.010		0.0547
rod elems. w-out offset	$[\Delta w'] / [\Delta w']_{tr} \%$	-5.289	-5.889	-0.513	-0.563	0.0331	0.0351
	$[\Delta \Theta] / [\Delta \Theta]_{tr} \%$	0.603	-0.096	-0.777	0.105	-0.1439	0.0222
	$[\Delta a^{el}] \%$		-5.984		-0.457		0.0573
add. othotr. layer w.-out offset	$[\Delta w'] / [\Delta w']_{tr} \%$	-3.851	-4.288				
	$[\Delta \Theta] / [\Delta \Theta]_{tr} \%$	-1.137	0.169	-		-	
	$[\Delta a^{el}] \%$		-4.119				
add. othotrop. layer + offset	$[\Delta w'] / [\Delta w']_{tr} \%$	0.647	0.724				
	$[\Delta \Theta] / [\Delta \Theta]_{tr} \%$	-0.086	0.011	-		-	
	$[\Delta a^{el}] \%$		0.735				
add. isotrop. layer + offset	$[\Delta w'] / [\Delta w']_{tr} \%$	-3.944	-4.383	-0.986	-1.075	0.0039	0.0042
	$[\Delta \Theta] / [\Delta \Theta]_{tr} \%$	-6.261	0.963	-1.968	0.267	0.7831	-0.1218
	$[\Delta a^{el}] \%$		-3.419		-0.807		-0.1176

Fig. 4 Deviations of structural deformations obtained for different levels of simplified stiffener modeling

Effects of Simplified Stringer Modeling

Explicit modeling of stringer stiffened top and bottom covers including property association for stiffening members is very time-consuming especially if the stringer cross-section geometry varies in both chord and span wise directions. Several degrees of stringer idealization are considered within the present study. A common method to avoid the modeling effort is to create a skin-stringer-"laminate" with isotropic skin and orthotropic stringer layers. The benefit of this approach is that only one parameter is required to realize the skin-stringer-structure. This parameter is the area ratio of the skin and summarized stringer cross-sections, commonly given in the literature as 100:50 for the design of transport aircraft wing structures [7]. The structural model with stringers smeared as an isotropic layer presents the simplest approach concerned in the present study.

Homogenizing discrete stringers over the skin area has two opposite effects on the bending stiffness of the wing box. The first effect is the reduction of the wing box local moment of inertia by neglecting the bending stiffness of the stringers. The second effect is the overestimation of the wing bending stiffness caused by neglecting the (offset) distance of the stringer cross-sections relative to the skin surface. The effect of ignoring the bending stiffness of the stiffeners on the bending and torsional deformation of the wing can be concerned on the basis of deviations obtained for a FE model with stringers realized with rod elements. For this idealization, the bending angle is increased by only 0.84% due to lower structural stiffness, resulting in 0.87% greater angle of attack. The influence of stringer stiffness on the torsional behavior and in turn on the elastic angle of attack is negligible (see table 4). The marginal impact of stringer bending stiffness on the deformation behavior results in change of geometric angle of attack being only 0.19% (see fig. 6).

In contrast to the effect of the stringer-stiffness, the overestimated contribution of the stringer-cross-sections to the local wing box moments of inertia due to non-considering the correct offset distance dominates the influence on the bending deformation. How can be seen in table 4 for the deviations obtained for FE models with stringers idealized as isotropic and orthotropic layer without offset, the bending angle decreases by 3.9 – 4%. Because stringers do not contribute to the shear load resistance of skin panels the stringer idealization as an orthotropic material layer enables to reproduce the torsional stiffness of the wing box structure in the way that

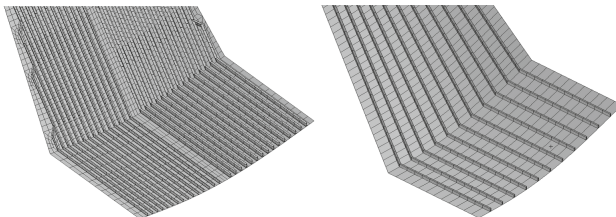


Fig. 5 Different levels of stringer modeling detail

is more realistic compared with stringers homogenized as isotropic material. This trend is demonstrated by the smaller deviation of twist ($[\Delta\Theta] = 1.1\%$) compared to the simplest model ($[\Delta\Theta] = 6.3\%$). One remarkable effect of greater difference of wing twist is the smaller deviation of elastic angle of attack of the structural model with stringers idealized as isotropic layer ($[\Delta\alpha^{el}] = -3.4\%$) compared with the more realistic approach ($[\Delta\alpha^{el}] = -4.1\%$). The trend predicted by the comparison of the $[\Delta\alpha^{el}]$ -deviation parameters between the both structural models, confirms with the results of static aeroelastic analysis. The deviation of converged angle of attack for the skin-stringer compound realized with orthotropic stringer layer is slightly higher ($\Delta\alpha_{EqSt} / \alpha_{EqSt} = -0.82\%$) as for a simpler structure ($\Delta\alpha_{EqSt} / \alpha_{EqSt} = -0.71\%$, see fig. 6).

Effect of Simplified Spar Cap and Rib Cap Modeling

Effects occurs by modeling the spar caps with beam or rod elements with or without considering element offsets are similar to those discussed in the section above. Due to smaller cross-section of the spar caps relative to the cross-section of the whole wing box this effects causes only marginal discrepancies of the bending and torsion angle and thus of the elastic angle of attack. If spar stiffeners are considered as isotropic layer in the wall thickness of the skin parts the bending stiffness increases causing 0.8% smaller elastic angle of attack. The deviation $[\Delta\alpha^{el}]$ that results from neglecting the element offsets varies between -0.46% and -0.54% (see fig. 4). A static aeroelastic response was calculated for wing structure with spar caps modeled with bar elements without element offset. The converged angle of attack

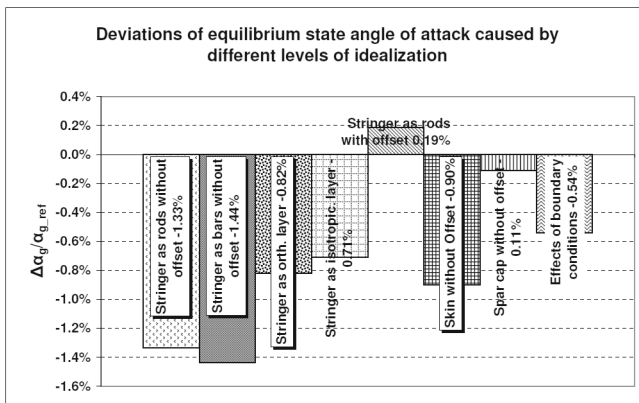


Fig. 6 Deviations of equilibrium state angle of attack for different variants of modeling simplification

α_{EqSt} of this structure is only 0.11% smaller compared to the reference structure. How can be seen from results in table 4 the influence of rib caps on the bending and torsional deformation is marginal resulting in deviations of elastic angle of attack being between 0.02% and 0.2%. Therefore, the influence of these stiffening components on the deformation behavior and thus on the aerodynamics of the wing can be neglected.

2.3.3 Conclusions

Within the first part of the work, a simple method was presented to calculate global parameters, which enables to estimate the effect of uncertainties of structural models on the deformation behavior and thus on the aerodynamic properties of the wing structure. This method was applied to investigate the impact of modeling uncertainty on the structural and aeroelastic response of the wing of a wide-body transport aircraft. The results of the study yield a rather good agreement between the deviation trends of the structure subjected to modeling uncertainty, which are calculated for a static loading and the discrepancy of aerodynamic properties of the wing obtained by a coupled analysis. As mentioned above, the elastic angle of attack α^{el} , employed as evaluation parameter is dominated by the bending deformation of the wing structure. Since the top and bottom covers have the greatest contribution to the bending stiffness of the wing, the simplified modeling of stringers has the major effect on the accuracy of the structural model. The deviations of converged angle of attack α_{EqSt} , used as performance criterion to evaluate the accuracy of the coupled analysis varies between 0.2% and 1.44% for different degrees of modeling detail (see fig. 6).

As shown on the sample of simplified boundary conditions, the higher deviations of twist and bending angle must not as well produce higher discrepancy of converged angle of attack. In fact the deviations has to be transformed in flight direction using the interrelationship give in equation (2) to estimate the resulting effect of the discrepancies within both deformations on the load distribution.

3 Part II: Stochastic Simulations

3.1 Introduction

In the second part, the effects of stochastic uncertainties on the accuracy of static aeroelastic analysis are investigated. A parametric finite element model (see section 1.1) is used to simulate the scatter of the structural input parameters expressed as Gaussian standard normal distribution. Coupled aeroelastic analysis is performed to obtain the deviation of the wing aerodynamics for a discrete distribution of the stochastic input parameters using a high order panel code. A first order reliability method is employed to calculate the probability of change of aerodynamic performance parameters due to the variation of structural stiffness properties. The

results of the stochastic analyses performed for a simple test case are presented and demonstrate robust behavior of a coupled aeroelastic system subjected by moderately arbitrary structural parameters.

3.2 First Order Reliability Method

In the present work, the probability of failure P_f of the wing structure is computed. It describes the probability that the structure does not to comply with the predefined requirements. Thus, the term failure has to be distinguished from other terms, like e.g. crash or disaster. Since the coupled fluid-structure analyses are very time consuming, the first order reliability method (FORM) was implemented to calculate the stochastic characteristics of the wing [10]. FORM introduces the reliability index β to describe the reliability of the structure. The main input to the method is the limit state function $G(\mathbf{X})$, where \mathbf{X} is the vector of stochastic variables that influence the structure. By definition, the limit state function is positive, if the structure fulfils its requirements. Negative values are returned, if at least one requirement is violated.

In order to generate unique results for every problem, the vector of stochastic variables is transformed into a vector of standard normal random variables \mathbf{X} . This leads to a limit state function $G(\mathbf{X}')$ which is analyzed using the FORM routine. The FORM is a gradient based optimization procedure which calculates the minimum distance β between the limit state function defined by $G(\mathbf{X}') = \mathbf{0}$ and the origin of the standard normal variable space spanned by the normalized stochastic variables.

At the beginning of the FORM algorithm, a $\beta_{initial}$ has to be estimated. The better the estimation of this initial value factor the fewer iterations are needed in the algorithm to get the final β . With the $\beta_{initial}$ and the limit state function value, all parameters are defined to start the main iteration of the FORM algorithm consisting of three main steps: (cp. Haldar, Mahadevan [11])

- Transformation of stochastic variables into standard normal variable space. In order to get unique results, all non-standard normal variables have to be transformed. For normal variables, a general conversion can be applied, for other variables, the Method of Rackwitz and Fiessler [12] has to be used.
- Generation of derivatives of the limit state function with respect to the standard normal variables. The coupled fluid-structure model can not be solved algebraically. Thus, the derivatives have to be estimated by finite differences in the neighbourhood of the design point.
- Calculation of the direction, where the steepest trend in the limit state function occurs and estimation of a new design point and the corresponding β value

This iteration is repeated until the limit state function value is zero and the β value converges. The resulting β value is then transferred to the fitness value calculation routine of the optimization.

3.3 Combination of the FORM-Routine with Fluid-Structure Interaction Code Library

To simulate an impact of variation of structural parameters on the aeroelastic response of the wing, the ifls-code-library was embedded into the routines performing the FORM algorithm. A NASTRAN input file of the finite element wing model was created with the ability to vary the structural properties during the stochastic process. Two input parameters are defined to be altered within the wing box structure: the thickness t of the thin-walled structural members and the Young's modulus E of the material. For stochastic input parameters a normal distribution is assumed. The shape of the normal distribution and, therefore, the extent of the deviation of the input parameters are characterized by the coefficient of variation (COV) $V = \sigma/\mu$. The COV is defined as a ratio of the standard deviation σ to the mean value μ . For a random variable with $V = 0.1$ the probability is 31.7% that the deviation of this variable exceeds $\pm 10\%$.

3.3.1 Definition of the Limit State Function

To apply the FORM analysis to the coupled fluid-structure problem a realistic failure criterion had to be defined to describe the performance of the simulated wing structure. For this kind of problem the random input is given by a variation of structural parameters. The change of the converged angle of attack α_{EqSt} of the aeroelastic equilibrium state (cp. section 1.2) was used to estimate the impact of random input parameters on the aerodynamic properties of the investigated wing model. The deviation $\Delta\alpha_{EqSt}/\alpha_{EqSt}$ can be considered in both positive and negative directions. The higher values of α_{EqSt} caused by a lower Young's modulus or by reduction in wall thickness, respectively, are assessed to be more critical than smaller ones, caused by a stiffer wing structure.

The probability of deviation of equilibrium state angle of attack is investigated for different values of $\Delta\alpha_{EqSt}/\alpha_{EqSt}$ varying between 0.4% and 1.0%. Each value corresponds to a limit state function in the normal variable space, which is defined as:

$$G(\mathbf{X}') = \Delta\alpha_{EqSt} - \Delta\alpha_{EqSt,req} \quad (4)$$

The term $\Delta\alpha_{EqSt,req}$ defines the highest permitted deviation of the converged angle of attack. For a discrete limit state function and a distribution of random parameters (characterized by the coefficient of variance) the FORM algorithm calculates a combination of these parameters for which the reliability index β becomes minimum. For the inversion of the argument, the probability of the aeroelastic response represented by the limit state function becomes maximal.

An exemplary problem for two random variables X'_1 and X'_1 with two limit state functions $G_1(\mathbf{X}')$ and $G_2(\mathbf{X}')$ is depicted in fig. 7. Corresponding to the definition of the reliability index β the probability of $G_1(\mathbf{X}')$ is higher than of $G_2(\mathbf{X}')$ because

of the smaller distance β between the curve and the origin of the standard normal space.

3.4 Sensitivity Analysis by a Global Variation in Structural Parameters

The variation of the wall thickness and Young's modulus causes a deviation of stiffness qualities of the wing structure. Due to manipulation of structural properties the tendency of the wing is affected to exceed its shape under a certain load. The objective of the parameter study was to estimate the impact of parameter variation within the main structural components on the structural behavior as well as on the static aeroelastic response.

The alteration of structural parameters of skin, spars, or ribs influences the torsion and bending distortions in different ways. Reduction of the wall thickness as well as of the Young's modulus in the skin parts has the highest effect on the bending and shear stiffness of the wing reducing the bending moment of inertia and shear coefficient of a local wing box cross-section. The torsional stiffness is also affected, depending on the ratio of wing box height to depth and thickness ratio of the skin to spar webs. Reduction of structural parameters in the spar webs influences mainly the torsional and shear stiffness having only a secondary effect on the bending moment of inertia. Due to the low contribution of the ribs to the bending and torsional

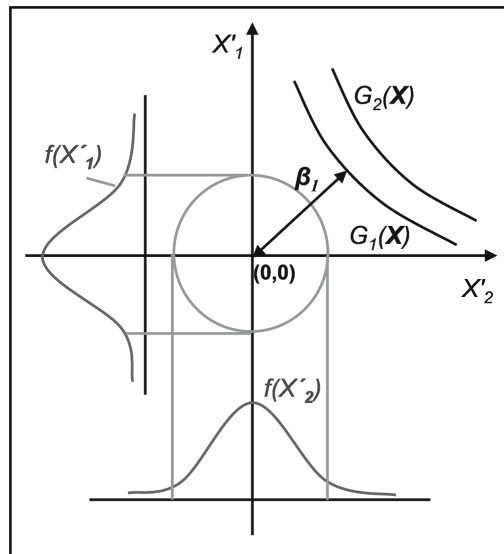


Fig. 7 Random input parameter distribution and limit state functions in the normal variable space

stiffness of the wing box structure the variation of the input parameters in this structural member has only a marginal effect on the deformation behavior of the wing.

A parameter study is carried out to estimate the sensitivity of the structural and thus of the static aeroelastic response relative to the components of the wing structure affected by uncertain input parameters. The influence of each component is estimated by changing successively the wall thickness and Young’s modulus of the skin, spar webs and ribs. To avoid local effects both input parameters are varied simultaneously by $\pm 10\%$ along the wing span. A structural and an aeroelastic response of a modified structure are determined for a reference loading corresponding to the 1g load case. From the structural response, the global deviations $[\Delta\Theta]$, $[\Delta w']$ and $[\Delta\alpha^{el}]$ of torsion deformation, bending angle and elastic angle of attack including the components $[\Delta\Theta_{tr}]$ and $[\Delta w'_{tr}]$ are calculated. An alteration $\Delta\alpha_{EqSt}/\alpha_{EqSt}$ of the converged angle of attack is obtained from the results of coupled analysis by comparison with the reference structure. The results for the global deviations are given in tables 4 and 5.

The wing box investigated in the parameter study which structural properties are varied separately and in the same manner does not represent a real wing. An actual wing structure is assembled of many different parts of which the dimensions and material properties vary independently from each other. The intent of this simple

Table 4 Deviations of bending angle, twist and elastic angle of attack caused by reduction of skin thickness by 10%

	skin		spars		ribs	
	rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.
$[\Delta w']/[\Delta w'_{tr}]/\%$	5.720	6.585	0.249	0.280	0.066	0.077
$[\Delta\Theta]/[\Delta\Theta_{tr}]/\%$	5.735	-1.172	-0.473	0.098	-0.306	0.063
$[\Delta\alpha^{el}]/\%$		5.412		0.377		0.140

Table 5 Deviations of bending angle, twist and elastic angle of attack caused by reduction of Young’s modulus by 10%

	skin		spars		ribs	
	rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.	rel. glob. deviation	rel. glob. deviation: transf.
$[\Delta w']/[\Delta w'_{tr}]/\%$	5.745	6.621	0.447	0.508	0.062	0.070
$[\Delta\Theta]/[\Delta\Theta_{tr}]/\%$	5.568	-1.138	-0.234	0.049	-0.279	0.058
$[\Delta\alpha^{el}]/\%$		5.483		0.556		0.128

approach is only to estimate the main trend of the deviation of the output parameters depending on the component of the structure in which the variation of input parameter occurs.

To estimate the tendency of change of the equilibrium state angle of attack α_{EqSt} caused by the input variation of structural components a static aeroelastic response is calculated for each modified structural model already described. The relative deviation of this angle is plotted in fig. 8 for each model derivate. The results show a good agreement with the tendencies obtained from the simple deformation study (see tables 4 and 5). The contribution of deviation of both deformation components to $\Delta\alpha_{EqSt} / \alpha_{EqSt}$ is somehow different for the variation of structural parameters in spar webs and rib surfaces. The change of the torsion angle is negative with respect to the sign convention showing therefore a stiffer torsional behavior. This tendency is due to the skewed root rib of a swept wing which influences the warping moment of inertia and, thus, the torsional behavior of the wing box.

A variation of the structural parameters shows the highest effect on the structure's stiffness and therewith on the change of the angle of attack in the skin areas as expected. The results of the structural response show that, in spite of a rather high ratio of the torsion angle to the elastic angle of attack, the latter is still dominated by the angle of bending deformation. The almost identical values obtained for $\Delta\alpha_{EqSt} / \alpha_{EqSt}$ by variation of both parameters of the skin parts should be treated as a special case taking into account the global character of the applied variations.

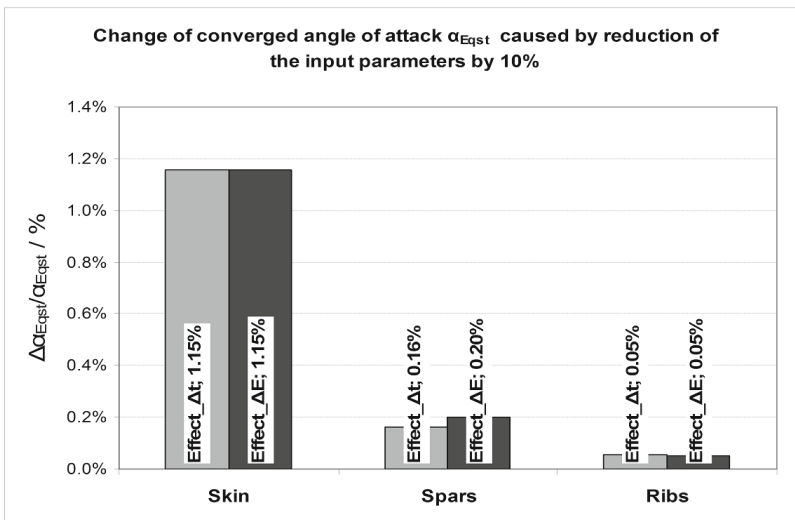


Fig. 8 Random input parameter distribution and limit state functions in the normal variable space

3.5 Results of the FORM Analysis

Within the stochastic analysis, the impact of random input parameters on the static aeroelastic response of the transport aircraft wing is investigated. Based on the results of the sensitivity analysis the analysis is performed at first only for skin areas due to the crucial impact on the wing aerodynamics. The wing structure is divided into four areas in which the input parameters were independently varied. The division of the areas is given in table 6 as a function of the span co-ordinate.

In each area, the structural parameters were varied simultaneously in the top and bottom skin parts. By this simplification, the number of random variables X'_i decreased to a total of four that in turn led to significant reduction of numerical expenditure.

The Gaussian normal distribution for random input parameters is assumed (see section 3.2). To estimate the coefficient of variance for the thickness distribution manufacturing data sheets for maximum thickness deviation were analyzed. Following this analysis, a coefficient of variance, which lies between 0.02 and 0.04, seems to be realistic, but the results were calculated until the $COV = 0.05$ showed the effect of greater scatter within the input parameters. For the variation of the Young's modulus, the same coefficients are used to guarantee the comparability of the results.

The allowed relative deviation $\Delta\alpha_{EqSt} / \alpha_{EqSt}$ of the converged angle of attack compared to the reference structure is analyzed in the range between 0.4% and 1.0% for different coefficients of variation. Each value of this deviation defines a limit state function $G(\mathbf{X}')$. For a given value of $G(\mathbf{X}')$ the FORM routine calculates a combination of random variables for which the reliability index converges. With regard to the investigated problem a combination of relative deviations of the structural input parameters in each area was found for which the probability of a given deviation of the angle of attack becomes a maximum.

The results of the variation in the wall thickness and the Young's modulus in skin areas are presented in figures 9 and 10. In the diagrams a probability of failure is plotted for a series of limit state functions over the coefficient of variance V . Due to almost linear correlation between the reliability index and limit state functions for a given COV, some curves could be extrapolated from the calculated results. These curves are plotted by dashed lines. For each limit state function the probability of the failure arises with the scatter in the input parameter expressed by V . The lower

Table 6 Areas of parameter variation

Area	η_i	η_0
1	0.0	0.22
2	0.22	0.44
3	0.44	0.72
4	0.72	1.0

the allowed difference of the angle of attack expressed by a failure function is the higher is the probability to violate the requirements.

In the diagrams a probability of failure is plotted for a series of limit state functions over the coefficient of variance V . Due to almost linear correlation between the reliability index and limit state functions for a given V , some curves could be extrapolated from the calculated results. These curves are plotted by dashed lines in fig. 5 and 6. For the investigated coefficients of variance the results for a relative deviation of an angle of attack of 1.5% were calculated for the local variation in the skin thickness of 10% and more. This degree of variation within the wing structure seems not very realistic to be considered further. For each limit state function the probability of the failure arises with the scatter in the input parameter expressed by V . The lower the allowed difference in the angle of attack, expressed by a failure function the higher is the probability to violate the requirements.

The comparison of the results for variation of skin thicknesses and Young’s modulus shows very similar probability curves for both input parameters. The probability of failure obtained for the variation of Young’s modulus is somewhat smaller as for a variation of skin thicknesses. This tendency shows a good agreement with the predictions made within the sensibility study carried out in section 3.4.

From the results of the stochastic analysis depicted in figures 9 and 10 it can be seen that the probability of higher deviations (<1%) of the global aerodynamic properties of the wing still are very small even for a higher variance of structural parameters. This demonstrates a high robustness of the coupled fluid structure system affected by the considered type of uncertainty.

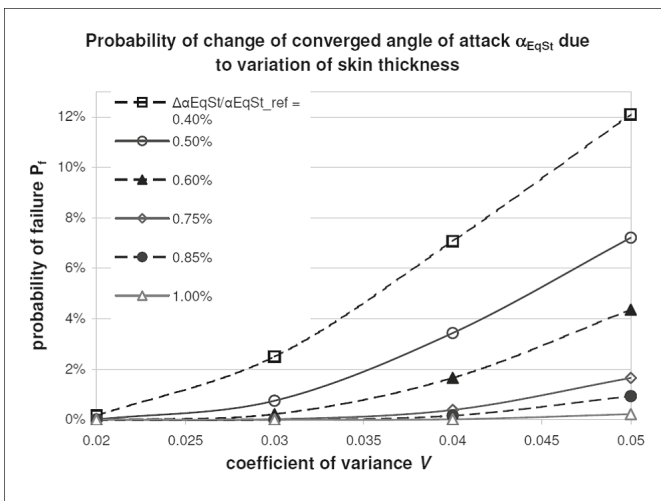


Fig. 9 Probability of deviation of angle of attack caused by variation of skin thickness for different performance criteria

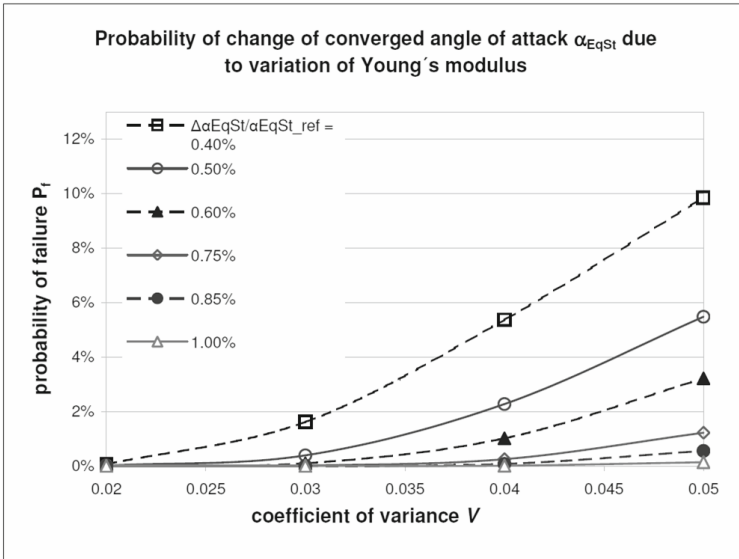


Fig. 10 Probability of deviation of angle of attack caused by variation of Youngt’s modulus for different performance criteria

3.6 Conclusions

In the present work, the influence of random structural parameters on the aerodynamic performance of a metallic test wing structure is investigated. The investigations demonstrate the suitability of the FORM analysis to handle some classes of stochastic uncertainties affecting the aeroelastic response of a wing structure. Due to the gradient based optimization procedure, which forms the basis of the FORM the main requirement to the investigated problem is the existence of only one minimum solution for the reliability index β . To handle problems which violate this requirement as there are the uncertainties of fibre orientation angles of composite materials, other stochastic analysis methods like Latin hypercube sampling should be used instead of the FORM.

To reduce the numeric costs of stochastic simulation some simplifications had to be made within the analysis process. The influence of the weight reduction on the target lift caused by reduction of the wall thicknesses was neglected. The simultaneous variation of structural parameters of the top and bottom skin in only four areas represents a highly simplified test case compared to the real structure (cp. the remarks in section 3.4). Considering these simplifications, the results obtained in the present work should represent a conservative trend.

The variation of the input parameters of top and bottom skin parts as well as of spar webs for a higher number of independent areas of variation is a part of actual work as well as the consideration of weight reduction for the target lift. Another effect which could be considered is the tendency of the skin areas to buckle if the

local bending stiffness of the panes is reduced by a variation of structural parameters having a significant influence on the aerodynamic drag.

References

- [1] Heinze, W.: Ein Beitrag zur quantitativen Analyse der technischen und wirtschaftlichen Auslegungsgrenzen verschiedener Flugzeugkonzepte für den Transport großer Lasten; ZLR-Forschungsbericht 94-01, Braunschweig (1994)
- [2] Österheld, C.M.: Physikalisch begründete Analyseverfahren im integrierten multidisziplinären Flugzeugvorentwurf; ZLR-Forschungsbericht 2003-06, Braunschweig (2003)
- [3] Reimer, L., Braun, C., Bae-Hong, C., Ballmann, J.: Computational Aeroelastic Design and Analysis of the HIRENASD Wind Tunnel Wing Model and Tests. In: International Forum on Aeroelasticity and Structural Dynamics (IFASD) 2007, Stockholm, Sweden, Paper IF-077 (2007)
- [4] Heinrich, R., Dargel, G.: Spezifikation des Testfalls für den Hauptmeilenstein M8.1 im Verbundvorhaben MEGADESIGN (2006)
- [5] Haupt, M., Niesner, R., Unger, R., Horst, P.: Coupling Techniques for Thermal and Mechanical Fluid-Structure-Interactions in Aeronautics. PAMM ü Proc. Appl. Math. Mech. 5, 19–22 (2005)
- [6] Schneider, W.: Die Entwicklung und Bewertung von Gewichtsabschätzungsformeln für den Flugzeugvorentwurf unter Zuhilfenahme von Methoden der mathematischen Statistik und Wahrscheinlichkeitsrechnung; Berlin, Techn. Univ., Diss. (1973)
- [7] Niu, M.C.Y.: Airframe Structural Design: practical Design Information and Data on Aircraft Structures. Conmilite Pr., Hong Kong (1999)
- [8] Patnaik, S., Gendy, A., Berke, L., Hopkins, D.: Modified Fully Utilized Design (MFUD) Method for Stress and Displacement Constraints. NASA Technical Memorandum 4743 (1997)
- [9] Malcolm, D.J., Laird, D.L.: Extraction of Equivalent Beam Properties from Blade Models. Wind Energy 10, 135–157 (2007)
- [10] Reim, A., Horst, P.: Structural optimization considering stochastic variations of manufacturing alternatives. In: 8th World Congress on Structural and Multidisciplinary Optimization (2009)
- [11] Haldar, A., Mahadevan, S.: Probability, Reliability and Statistical Methods in Engineering Design. Wiley & Sons (2000)
- [12] Rackwitz, R., Fiessler, B.: Structural Reliability Under Combined load sequences. Computers & Structures 9, 489–494

A Comparison of Fluid/Structure Coupling Methods for Reduced Structural Models

Georg Wellmer, Lars Reimer, Horst Flister, Marek Behr, and Josef Ballmann

Abstract. In this paper, the realisation and testing of spatial coupling methods for aeroelastic simulations with partitioned algorithms is presented. The investigated methods for spatial coupling—the transfer of loads and deformations between the wetted surface and the structural model—are the method of Finite Interpolation Elements and two other, newly-implemented interpolation methods. All three are suitable for reduced structural models, and the geometries of the wetted surface and the structural model do not have to coincide. The aeroelastic simulation tool employed and the theoretical background of the spatial coupling schemes are outlined. Different measures for the quality of the spatial coupling are derived and applied to test cases of increasing complexity. The influence of user-defined coupling parameters on the deformation projection is assessed. Based on these results and on practical considerations, the available coupling methods are compared and conclusions are drawn regarding their applicability.

1 Introduction

The civilian aircraft industry faces the necessity to reduce aircraft fuel consumption while increasing flight safety levels and maintaining passenger comfort. Furthermore, competition on the aircraft market forces manufacturers to accelerate design cycles and to reduce the costs of the actual development. This twofold pressure has brought about the widespread adoption of numerical prediction methods during all

Marek Behr · Georg Wellmer · Lars Reimer

Chair for Computational Analysis of Technical Systems (CATS), Center for Computational Engineering Science (CCES), RWTH Aachen University, Schinkelstraße 2, 52062 Aachen
e-mail: [wellmer, reimer, behr}@cats.rwth-aachen.de](mailto:{wellmer, reimer, behr}@cats.rwth-aachen.de)

Josef Ballmann

Lehr- und Forschungsgebiet für Mechanik (LFM), RWTH Aachen University,
Schinkelstraße 2, 52062 Aachen
e-mail: ballmann@lufmech.rwth-aachen.de

stages of the design process. Computational Fluid Dynamics (CFD) for the numerical prediction of the flow field about aircraft configurations are of special interest to the industry. These methods have matured to a point where they are not merely complementing costly wind tunnel test campaigns, but actually partially supplanting them. Simultaneously, improvements in structural analysis methods such as Computational Structural Dynamics (CSD) and in material sciences have led to lighter aircraft frames with greater inherent elasticity. Aeroelastic coupling effects now definitely have to be considered in the design process and thus also have to be captured by the numerical prediction methods, i.e. by Computational Aeroelasticity (CAE) solvers. A code package for the simulation of the interaction between aerodynamic, elastic and inertial forces has been developed at LFM/CATS over the last decade. Work was initiated at LFM within the framework of the Collaborative Research Centre 401 (SFB 401) [3, 28] and continued at CATS, in the course of the collaborative research project MUNA, amongst others.

In order for CAE to gain the same acceptance in the aerospace community as that already enjoyed by CFD, its solutions must prove to be trustworthy. Engineers require the numerical predictions design decisions are based on to have a dependable accuracy, which can be evaluated in two different manners: First of all, by comparison with experimental results the error incurred by the whole coupled algorithm can be estimated. It can then be used as a measure of confidence for numerical predictions regarding comparable configurations. Validation against steady and unsteady wind tunnel data has been carried out continuously at LFM/CATS, most recently in the project “High Reynolds Number Aero-Structural Dynamics” (HIRENASD) [4, 24]. This approach has the downside that without extensive parameter studies the cause of deviations—potentially each of the single-field

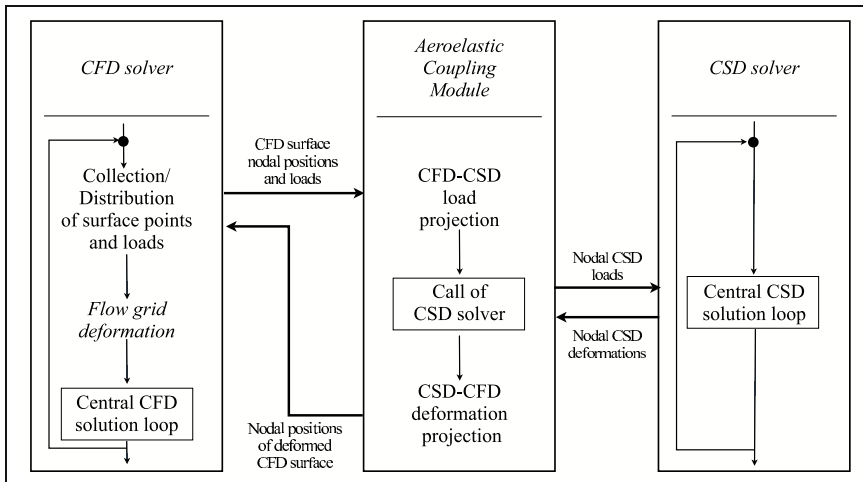


Fig. 1 General concept of the ACM and its data exchange with the CFD and CSD solvers (taken from [28])

solvers or their mutual interaction—cannot be easily determined. Besides, it must not be forgotten that also measurements inevitably have an error. In the second approach error sources are identified and examined individually, at least as far as such a separation is possible. Regarding aeroelastic coupling, it has to be demonstrated that associated sources of error do not significantly impair the accuracy of the overall coupled solution. This method often is feasible for model test cases only, and the findings have to be scrutinised before being applied to real-world problems. During the MUNA project both outlined investigation methods have been applied at LFM/CATS with regard to steady aeroelastic simulations and their associated error sources.

This paper will take the following outline: First the coupling methodology in general and the algorithm developed at LFM/CATS in particular are delineated, as well as the available spatial coupling methods. The potential error sources are defined and their influence is quantified for model problems. Next, the assessment is repeated for actual coupled flow simulations. Based on the results, the available methods for spatial coupling are compared and a set of recommendations is derived.

2 Coupling Methodology

All algorithms for the simulation of fluid-structure interaction problems fall into one of two major categories: Monolithic algorithms solve the equations governing the flow field and those governing the structural deformation simultaneously as a single set of equations [7, 21]. Partitioned algorithms employ dedicated solvers for each field which are coupled via a suitable interface. The monolithic method ensures that the mutually dependent solutions in each field are always on the same time level, which eliminates the issue of synchronising individual solvers for a conservative solution. In practise, though, this method has one significant disadvantage, which has limited its acceptance: A monolithic coupled solver generally has to be developed completely from scratch, whereas with a partitioned approach one can employ pre-existing single-field solvers and benefit from the developments of specialised research groups. Ideally, the necessary coupling interface should be sufficiently modular to allow the replacement of a single-field solver either with an updated version or with an entirely different implementation. The aeroelastic code package conceived at LFM/CATS, henceforth denoted as “Aeroelastic Coupling Module” (ACM) [9, 24, 25], is based on this rationale. The ACM allows the modular coupling of arbitrary flow and structural solvers with only minor code changes. Both steady simulations with a staggered (Block-Gauss-Seidel) algorithm and unsteady simulations with weak and strong temporal coupling schemes are possible. Pursuant to the partitioned approach followed here, the ACM serves as the interface between the dedicated single-field solvers for the flow field and for the structural deformation, as is shown in Fig. 1. The ACM carries out the synchronisation of the solvers by initiating iteratively their respective calls.

Apart from the synchronisation of the single-field solvers, the ACM also performs the spatial coupling, i.e. the projection of loads from the wetted surface to the structure and in reverse direction the projection of the structural deformations to the wetted surface. The projection methods available are tailored to reduced structural models. These are beneficial especially during unsteady simulations because of their smaller number of degrees of freedom and thus lower requirements of computational resources. With reduced structural models the geometries of the wetted surface and of the structural model coincide only in parts or not at all. This is especially true for beam models which do not even share the same dimensionality as the wetted surface. Also with more detailed models like shell models, in many cases one does not want to represent the complete structure. When modelling a wing, often only the wing box is taken into consideration. The high lift devices and other components which do not contribute significantly to the overall structural stiffness are disregarded. In both examples there are “gaps” between the wetted surface and the structural model which have to be bridged by the projection algorithm.

Because of these gaps between wetted surface and structure, forces have to be projected from the wetted surface to the structure instead of surface stresses. Latter do not possess an effective direction required in this case. As to increase the modularity of the ACM, the aerodynamic surface forces are calculated already inside the flow solver and passed on to the ACM. They should be derived in a consistent manner from the discrete distribution of surface stresses [12]. The ACM receives a cloud of load incidence points with associated force vectors and returns a cloud of surface coordinates representing the deformed wetted surface. Thus, the ACM can be coupled with structured and unstructured flow solvers alike since it is independent of the manner in which points are associated with surface cells. As a side note, the number of the load incidence points and their position in the undeformed wetted surface do not have to be identical with the surface points defining the wetted surface.

To summarise above statements, the aeroelastic coupling with the ACM comprises three steps:

1. From the pressure and surface stress distribution on the wetted surface, discrete force vectors are determined by the flow solver.
2. These surface forces are projected from the wetted surface to the nodes of the structural model.
3. The structural deformations resulting from the projected load distribution are projected back onto the wetted surface.

Each of these steps may contribute to the total error of the coupled simulation scheme. The second and third ones are closely related, though: The projection of (generalised) forces from the wetted surface to the structure can conveniently be expressed as a matrix-vector product with a force projection matrix P_F , and likewise the projection of (generalised) deformations can be expressed with a deformation projection matrix P_U :

$$\mathbf{F}_{\text{CSD}} = P_F \mathbf{F}_{\text{CFD}} \quad \text{and} \quad \mathbf{u}_{\text{CFD}} = P_U \mathbf{u}_{\text{CSD}} \quad (1)$$

The conservativity of the projection method is assured if $P_U = P_F^T$, which can be shown via the principle of virtual work [12]:

$$\begin{aligned}
 \delta W_{CFD} &= \mathbf{F}_{CFD}^T \delta \mathbf{u}_{CFD} & \delta W_{CSD} &= \mathbf{F}_{CSD}^T \delta \mathbf{u}_{CSD} \\
 &= \mathbf{F}_{CFD}^T P_U \delta \mathbf{u}_{CSD} & &= (P_F \mathbf{F}_{CFD})^T \delta \mathbf{u}_{CSD} \\
 & & &= \mathbf{F}_{CFD}^T P_F^T \delta \mathbf{u}_{CSD}
 \end{aligned} \tag{2}$$

Consequently, the same projection method has to be used during the projection of forces as during the projection of deformations.

A fourth step, external to the ACM, involves the deformation of the CFD volume mesh in order to accommodate the deformed wetted surface. Mesh deformation methods generally depend on the formulation of the flow solver employed, and their associated error sources are not investigated here.

2.1 Flow Solver

To date, the ACM has been coupled with three Reynolds-Averaged Navier-Stokes (RANS) Finite-Volume flow solvers: FLOWer [18, 19], TAU (recent developments are highlighted in a number of papers in this volume) and QUADFLOW [3]. In this paper, results obtained with FLOWer and with TAU are presented. The development of both solvers was initiated and led by the DLR. Further enhancement of the structured solver for multi-block topologies—FLOWer— may not be actively promoted anymore, but with that solver the greatest amount of experience has been gained in conjunction with the ACM, and the coupling can be regarded as well-validated against experiments. The effort to couple the ACM with the hybrid-unstructured solver TAU began during the previous project MEGADESIGN [20] and, for the steady branch, has been completed during MUNA. The interfaces of both flow solvers with the ACM provide the same functionality, but their implementation is quite different. FLOWer simply calls the ACM as a Fortran subroutine. The loads and load incidence points and the coordinates of the deformed wetted surface are exchanged via the subroutine parameter field. The communication between TAU and ACM used to be realised by files written on hard disk, but is now carried out completely in-memory. The solution is controlled through a script written in the object-oriented scripting language Python [26]. TAU and its components already have Python interfaces and are suitably wrapped during compilation. The ACM has to be provided as a shared object file. Specific Python interface classes contain the methods and attributes needed to perform the aeroelastic coupling. In each coupling step, the ACM’s Python interface reads the loads and load incidence points from TAU’s C data stream and passes them on to the ACM. In the reverse direction at the end of a coupling iteration, the interface receives the coordinates of the deformed wetted surface from the ACM and writes these to the data stream. All the while the interface has to ensure that the fields are passed on correctly between the individual software components written either in C, Fortran or Python.

For an investigation of the actual projection process it may not be relevant whether or not the loads on the wetted surface result from a flow simulation. In certain cases a user-defined load distribution may be imposed instead. This is possible with the stand-alone version of the ACM. Obviously, conclusions drawn in this manner can only regard the projection algorithm as such and not the coupled solution process as a whole.

2.2 *Structural Solver*

For the computation of the structural deformation, the in-house structural solver “Finite Element Analysis for Aeroelasticity” (FEAFA) is employed. It is a Finite Element (FE) code based on a physically and geometrically linearised formulation, so it is limited to small strains and linear-elastic material behaviour. Over recent years, it has been expanded to offer a range of element types comparable to commercial CSD code packages which includes volume and shell elements, spring elements, point masses and multi-freedom constraints. The mainstay for aeroelastic simulations is the multi-axial Timoshenko beam element [8, 9]. Its formulation allows for distinct cross-sectional positions of the centre of mass, the shear centre and the centre of bending. Thereby structural coupling between bending and torsional motion can be captured. The consideration of shear deformation in the Timoshenko formulations assures a physically-reasonable wave propagation through the structure, which is important for unsteady simulations. With very few degrees of freedom and thus at low computational cost, such reduced structural models are capable of accurately rendering the elastic and inertial properties of slender structures such as transport aircraft wings. This is not only a significant advantage for unsteady simulations, but also for steady design optimisation tasks, as has been demonstrated in the MEGADESIGN [20] project. During steady simulations, the structural deformation is either obtained by direct solution of the linear system of equations resulting from the FE discretisation or by superposition of pre-calculated modes.

2.3 *Flow Grid Deformation*

For the volume mesh deformation of structured FLOWer meshes, the Multiblock Grid Deformation Tool (MUGRIDO) [8, 16] was developed at LFM. This tool models the block boundaries of the volume mesh and selected additional mesh lines as massless Timoshenko beams. The deflections of the surface nodes relative to the undeformed configuration are imposed as boundary conditions and the structural problem is solved. Finally, the positions of the remaining mesh points inside of the blocks are calculated with transfinite interpolation. MUGRIDO is not suitable for unstructured TAU meshes. TAU offers two mesh deformation algorithms;

best suited for aeroelastic simulations with complex configurations is the weighted volume spline interpolation algorithm [15]. A further description can be found in the paper by Barnewitz in this volume. Since this method does not require any information regarding the connectivity between volume mesh points, it is equally applicable to structured meshes.

3 Spatial Coupling

The analyses presented in the paper at hand concentrate on error sources in spatial coupling, and so the description of the projection methods shall be afforded a separate section here. To begin with, an overview of projection methods suitable for reduced structural models is given. The existing projection method based on Finite Interpolation Elements (FIE) is explained. Then the newly-implemented Global Spline-Based (GSB) and Moving Least-Squares (MLS) methods are presented in detail.

In order to be valid from the physical point of view, any projection scheme has to be conservative with the following two criteria: First of all, the total force and moment vectors must be preserved during the projection. Secondly, during steady simulations the elastic strain energy of the structure must be identical to the work performed by the aerodynamic loads on the wetted surface, as implied by Eq. (2). During unsteady simulations also the instantaneous power exchanged over the coupling surface must be the same on both sides. From the flow solver and the volume mesh deformation code, further numerical requirements arise affecting the projection of deformations from the structure back to the wetted surface: The resulting deformed surface mesh should be contiguous in particular at intersections between the surface meshes of distinct assemblies, for instance between fuselage and wing. The deformed surface mesh should be smooth in order to assure good convergence of the flow solution. One final demand is of a more practical nature: With reduced structural models, any projection scheme has to make some kind of assumption for the transfer of forces and deformations over the gap between wetted surface and structure. This assumption should not be far removed from the load paths actually to be expected, i.e. some measure of locality should be preserved during the projection.

Initially, only one projection algorithm was available inside the ACM: The Finite Interpolation Element (FIE) method [5, 8, 9, 25] is an uncomplicated method that uses the shape functions of the structural model to divide aerodynamic surface loads among the nodes of the closest structural element. During the first phase of MUNA, a number of alternative projection methods was reviewed for inclusion in the ACM. Many published methods are only adequate for configurations where the wetted surface and the surface of the structural model coincide up to the discretisation error [10, 17, 22, 29]. On account of the requirements set forth for reduced

structural models such methods were excluded. Prospective methods included the Infinite-Plate Spline (IPS) method [14], the Constant-Volume-Tetrahedron (CVT) method [2, 13] and the inverse Boundary Element Method (BEM) [11]. These were extensively compared by Sadeghi et al. [27]. The GSB method [6] and the MLS method [23] constitute further alternatives suitable for reduced structural models.

With the IPS method, only the deflections normal to the wing plan form are interpolated from the structure to the wetted surface using splines as interpolation functions. This limits the method to (almost) planar configurations. With the CVT method, tetrahedra are spanned between the points on the wetted surface and the nodes of the closest structural element. Both the natural coordinates of the projection point inside the element and the volume of the tetrahedron are kept constant for all deformation states, defining the projection. As will become evident further down, the CVT method can be regarded as an extension of the FIE method. The BEM method is the projection method which is most firmly footed on physical considerations instead of geometrical neighbourhood relations: The gap between wetted surface and structure is modelled as an elastic continuum, and the deformation of the structure is expressed in terms of surface deflections through the BEM. This relation then has to be inverted with the minimisation of the elastic strain energy of the continuum as an additional constraint. Of all methods presented so far this is the most demanding. Furthermore, it requires the connectivity of the wetted surface, which currently is not transmitted from the flow solver to the ACM. The GSB and MLS methods both determine a function approximation to the nodal displacement distribution and evaluate it at the surface points. The two methods differ primarily in their choice of interpolation functions. They do not interfere with the modular structure of the ACM, offer the required generality, are independent of the dimensionality of the structural model and involve only a moderate implementation effort. Also, there is a significant implementation overlap between them, for which reason both were selected for inclusion in the ACM.

3.1 Finite Interpolation Element Method

The FIE projection method, also known as inverse isoparametric mapping, uses the shape functions of the FE structural model to interpolate loads and deformations between the points on the wetted surface and the nodes of the structural model. This results in an efficient algorithm which only requires the evaluation of algebraic expressions. The FIE method is briefly demonstrated here in conjunction with beam models. For a more elaborate description extended to structural models consisting of volume and shell elements the reader may refer to Reimer et al. [25].

The FIE method is based on purely geometrical considerations. In the first step, the closest structural element for a given point on the wetted surface is sought. Inside this element the projection point is determined. For a beam model this point

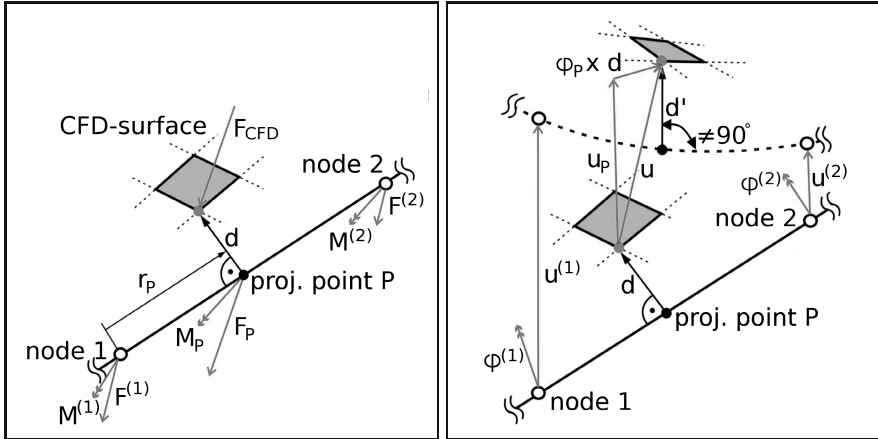


Fig. 2 Load and deformation projection with FIE on beam elements. *left*: Projection of aerodynamic forces from the wetted surface to the structure. *right*: Projection of deformations from the structure back to the wetted surface.

generally creates a perpendicular connection between surface point and beam axis, as depicted on the left of Fig. 2. The aerodynamic surface load \mathbf{F}_{CFD} is shifted along the distance vector \mathbf{d} to the projection point \mathbf{P} and an equivalent offset moment $\mathbf{M}_P = \mathbf{d} \times \mathbf{F}_{CFD}$ is introduced. With the shape functions of the element and the natural coordinate of the projection point r_P the force and the moment are divided among the element’s nodes. The closest elements and natural coordinates of the projection points are determined only once before the first coupling step and then reused. During the deformation projection shown on the right of Fig. 2 the corresponding steps are carried out in opposite order: The rotational deformation φ_P and the translational deformation \mathbf{u}_P at the projection point are interpolated from the nodal values with the element shape functions. The deflection of the surface point consists of \mathbf{u}_P and a rotational contribution $\varphi_P \times \mathbf{d}$. Even if the projection point on the undeformed beam axis created a perpendicular connection between beam axis and surface point, due to shear this may not be the case in the deformed configuration, represented by the dashed line in Fig. 2.

The methodology applied for structural models comprising shell or volume elements, which have two-dimensional projection surfaces, is the same in principle. A more involved algorithm is required to find the projection point on the closest element face. Interpolation in intersection regions has not been realised yet, so that with such structural models the FIE method in the ACM can be applied only to configurations with one assembly. The CVT method can be seen as a variant of the FIE method in which the length of the distance vector is no longer kept constant, but adapted according to the deformational change of the area of the projection face.

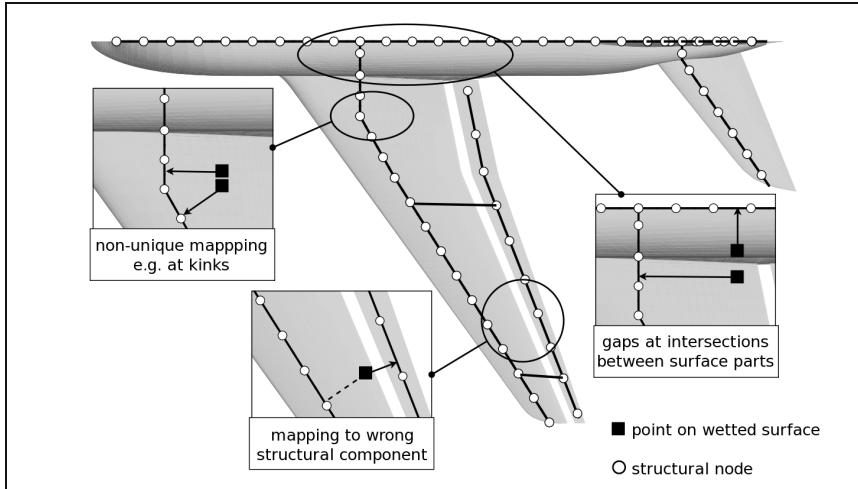


Fig. 3 Regions in which a straightforward application of the FIE method will lead to undesirable results.

3.2 Additional Interpolation Schemes for FIE

For structural models with a single straight beam axis, the FIE method is a logical extension of beam theory. The projection algorithm assures that during deformation sections through the wetted surface that were perpendicular to the beam axis in the undeformed configuration preserve their shape. If however a structure consists of several angled beam segments, possibly part of different assemblies such as a wing and fuselage, a straightforward application of the described algorithm can lead to a non-smooth or non-contiguous deformed wetted surface.

With configurations comprising more than one assembly, any projection between surface components and structural model parts not physically connected must be avoided. This is exemplified in Fig. 3 for a high-lift device. The structural elements closest to surface points along the trailing edge actually belong to the flap. Projections based solely on shortest distance lead to a physically impossible transfer of loads and deformations over the flap gap. This is prevented by explicitly assigning structural elements to surface segments of the individual assemblies. In a preparatory step all collinear beam elements of each assembly are combined in element groups. Each assembly's surface segment is given a unique identifier. In the ACM's input data set, the element groups are then either assigned to surface segments or excluded from the projection algorithm. In the current example, the flap track elements should be excluded because they have no wetted surface segments as counterparts for mapping.

However, the strict application of this explicit assigning can make the wetted surface come apart at intersections between assemblies: Due to their projection on

elements of different element groups, those neighbouring surface points which are part of different assemblies can experience incompatible deflections. The two surface segments necessarily contiguous in the undeformed configuration are no longer so after the deformation projection. The resulting defective mesh is not suitable any more for flow computations. This problem is resolved by means of an interpolation algorithm, which is exemplified by the wing-fuselage joint shown in the left image of Fig. 4. First, all seam curves between adjacent surface segments are detected. For a surface point belonging e.g. to the main wing, the projection is carried out onto the directly assigned element groups of the wing, giving a “direct” deflection $\mathbf{u}_{\text{CFD}}^{\text{dir}}$. Next, the projection is repeated for the element groups assigned to the neighbouring fuselage surface, which gives an “indirect” deflection $\mathbf{u}_{\text{CFD}}^{\text{indir}}$. The weighted average of the two contributions is taken:

$$\mathbf{u}^{\text{CFD}} = \frac{1}{1 + w_a} \mathbf{u}_{\text{CFD}}^{\text{dir}} + \frac{w_a}{1 + w_a} \mathbf{u}_{\text{CFD}}^{\text{indir}} \quad \text{with } w_a = w\left(\frac{a}{a_{\text{limit}}}\right). \quad (3)$$

The distance a of the surface point from the seam curve normalised with the user-supplied width of the intersection region a_{limit} determines the interpolation weight. The weighting function w is a high-order polynomial. The interpolation assures a smooth deformed wetted surface where neighbouring points on opposite sides of a seam curve have compatible deflections. A comparable algorithm has been realised by Badcock et al. [2] inside the CVT method.

In the concave region enclosed by the beam kink valid projections on multiple beam elements are possible. If for each surface point projection on the closest element is applied exclusively, some adjacent surface points will be paired with projection points far apart from each other. Their differing deformation values will result in creases in the wetted surface, which are neither physically justified nor favourable for the flow solver. This problem has to be resolved with an additional interpolation algorithm, which is explained along the beam model with two kinks shown on the right of Fig. 4. For a given surface point, a projection is determined onto each element group. For each one, the surface point deflection $\mathbf{u}_{\text{CFD},i}$ resulting from the structural deformation at the given projection point \mathbf{P}_i is calculated and two interpolation weights are assigned. The first interpolation weight w_β is a function of the deviation β of the projection angles from a right angle. The second interpolation weight w_d is defined by the ratio of distances of the projection points to the surface point:

$$w_{\beta,i} = w\left(\frac{\beta_i}{\beta_{\text{limit}}}\right), \quad w_{d,i} = w\left(\frac{d_i/d_{\text{min}}}{d_{\text{limit}}}\right) \quad (4)$$

β_{limit} and d_{limit} are user-defined parameters which determine the extent of the interpolation region. The final surface point deflection is then interpolated from all considered projection results:

$$\mathbf{u}_{\text{CFD}} = \sum_i w_{\text{tot},i} \mathbf{u}_{\text{CFD},i} \quad \text{with} \quad w_{\text{tot},i} = \frac{w_{\beta,i} w_{d,i}}{\sum_j w_{\beta,j} w_{d,j}} \quad (5)$$

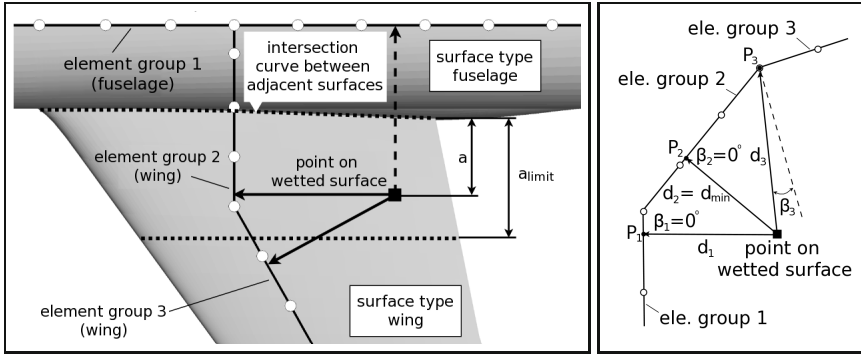


Fig. 4 *left*: Blending in the vicinity of intersections between surface segments of different assemblies, here at the wing-fuselage joint. *right*: Interpolation regions with non-unique mapping near kinks of the beam axis.

All interpolations have to be applied in the same manner also during the load projection as otherwise the conservativity would be violated (cf. Eq. (2)).

3.3 Global Spline-Based Interpolation (GSB)

As alternatives to the existing FIE scheme, in the project MUNA, the spatial coupling schemes GSB and MLS were implemented in the ACM. They are closely related, as both cast the problem of projecting loads or deformations as an interpolation problem: For a set of N points in space $\bar{\mathbf{x}}_n$ with dependent values $f(\bar{\mathbf{x}}_n)$ one seeks to find a functional approximation to f based on a suitable choice of interpolation functions. In this sense, the distribution of dependent values is the deformation \mathbf{u} provided at the structural nodes. Its functional approximation is then evaluated at a second set of M points $\hat{\mathbf{x}}_m$, which are the points of the CFD surface mesh. The two projection methods differ in their choice of interpolation functions which dictates the solution process.

The GSB method was originally published by Beckert and Wendland [6]. The authors approximate the deformation on the whole domain with a global low-order polynomial with Q monomials. The monomial vectors are either

$$\mathbf{m} = (1, x, y, z)^T \quad \text{or} \quad \mathbf{m} = (1, x, y, z, x^2, y^2, z^2, xy, yz, zx)^T. \quad (6)$$

Superimposed are local contributions $\phi(\mathbf{x})$ that consist of radial basis functions (RBF). At a given coordinate, the interpolation function is

$$s(\mathbf{x}) = \mathbf{m}^T(\mathbf{x})\boldsymbol{\beta} + \sum_{n=1}^{N_\delta} \alpha(\bar{\mathbf{x}}_n) \phi(\mathbf{x}, \bar{\mathbf{x}}_n). \quad (7)$$

The coefficients $\alpha(\bar{\mathbf{x}}_n)$ of the local RBF contributions and the coefficient vectors $\boldsymbol{\beta}$ of the global polynomial are calculated simultaneously for all N_δ support points with a weighted least-squares algorithm. The dependent values at the interpolation support points are reproduced exactly. The RBFs with compact support constructed by Wendland [30] serve as weighting functions. The C^2 -continuous Wendland-RBF with a support radius δ is provided here as an example:

$$\phi(\mathbf{x}, \bar{\mathbf{x}}_n) = (1 - \tilde{x})_+^4 (4\tilde{x} + 1) \quad \text{with} \quad \tilde{x} = \frac{1}{\delta} \|\mathbf{x} - \bar{\mathbf{x}}_n\|_2. \quad (8)$$

The index + marks that the factor $(1 - \tilde{x})^4$ is set to zero for values of $\tilde{x} > 1$, whereby the compact support is realised. The functional approximation to the deformation distribution can be obtained from the linear system of equations

$$\underbrace{\begin{bmatrix} [\phi(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}_j)] & [\mathbf{m}^T(\bar{\mathbf{x}}_i)] \\ [\mathbf{m}(\bar{\mathbf{x}}_j)] & 0 \end{bmatrix}}_{=C} \begin{Bmatrix} \{\alpha_i\} \\ \boldsymbol{\beta} \end{Bmatrix} = \begin{Bmatrix} \{u_\lambda(\bar{\mathbf{x}}_j)\} \\ \mathbf{0} \end{Bmatrix}, \quad 1 \leq i, j \leq N_\delta, \quad (9)$$

which has to be solved for each Cartesian displacement component $\lambda = x, y, z$. (Here, scalar quantities that are combined to a vector are put in braces. Brackets denote that scalars or vectors that are assembled to form a matrix.) This process would have to be repeated in each coupling step; instead the inverse of the coefficients matrix C is determined. The functional approximation can now be evaluated at the surface points, which yields the final projection matrix P:

$$\begin{Bmatrix} \{u_\lambda(\hat{\mathbf{x}}_j)\} \\ \mathbf{0} \end{Bmatrix} = \underbrace{[[\phi(\hat{\mathbf{x}}_i, \bar{\mathbf{x}}_j)] \quad [\mathbf{m}^T(\hat{\mathbf{x}}_i)]]}_{\text{P}} C^{-1} \begin{Bmatrix} \{u_\lambda(\bar{\mathbf{x}}_i)\} \\ \mathbf{0} \end{Bmatrix}, \quad (10)$$

$$1 \leq i \leq N_\delta \quad 1 \leq j \leq n_{\text{CFD}}.$$

In the GSB method the support radius δ has to be the same all over the computational domain, or else the interpolation scheme will not be consistent. In general, the number of support points N_δ will differ from one CFD surface node to the next. The user has to define a minimum required number of support points and the projection scheme searches the domain for the smallest radius δ that contains this number. Because of the global contribution to the interpolation function the resulting projection matrix is dense. Its definition here (and in the MLS method) differs from the definition in Eq. (2) in that here the deformations or forces are projected one spatial component λ at a time. The GSB method is largely identical to the volume mesh deformation method presented by Barnewitz in this volume. The main difference lies in the choice of weighting functions and the compact support of the local contributions to the interpolation function.

3.4 Moving-Least-Squares Interpolation (MLS)

The MLS interpolation method was first applied to spatial coupling in aeroelasticity by Quaranta et al. [23]. It exclusively uses low-order polynomials

$$s(\mathbf{x}) = \mathbf{m}^T(\mathbf{x}) \boldsymbol{\alpha} \quad (11)$$

to approximate the spatial deformation distribution $\mathbf{u}(\bar{\mathbf{x}})$. At each CFD surface point $\hat{\mathbf{x}}$, a new set of Q polynomial coefficients $\boldsymbol{\alpha}(\hat{\mathbf{x}})$ is computed with a moving least-squares fit. The N_δ closest support points $\bar{\mathbf{x}}_n$ provide a compact support; their influence relative to the CFD surface point is weighted with Wendland RBF [8]. For each surface point $\hat{\mathbf{x}}$ and its N_δ support points inside the support radius δ a functional

$$I(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) = \int_{\Omega_\delta} \phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) (\mathbf{m}^T(\bar{\mathbf{x}}_n) \boldsymbol{\alpha}(\hat{\mathbf{x}}) - u_\lambda(\bar{\mathbf{x}}_n))^2 d\Omega(\bar{\mathbf{x}}_n) \quad (12)$$

has to be minimised for the coefficients $\boldsymbol{\alpha}(\hat{\mathbf{x}})$. The discrete form of this functional is reduced to the normal equation through a variation of coefficients $\delta\boldsymbol{\alpha}$:

$$\underbrace{[\mathbf{m}(\bar{\mathbf{x}}_n)] \Phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) [\mathbf{m}(\bar{\mathbf{x}}_n)]^T}_{=A} \boldsymbol{\alpha}(\hat{\mathbf{x}}) = \underbrace{[\mathbf{m}(\bar{\mathbf{x}}_n)] \Phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n)}_{=B} \{u_\lambda(\bar{\mathbf{x}}_n)\} \quad (13)$$

Herein, $\Phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) = \mathbb{E} \{ \phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) \}$ is the diagonal matrix of RBF weighting factors. Inserting the interpolation function [11] yields

$$u_\lambda(\hat{\mathbf{x}}) = \underbrace{\mathbf{m}^T(\hat{\mathbf{x}}) \mathbf{A}^{-1} \mathbf{B}}_{=\mathbf{P}(\hat{\mathbf{x}})} \{u_\lambda(\bar{\mathbf{x}}_n)\}. \quad (14)$$

The row matrix $\mathbf{P}(\hat{\mathbf{x}})$ describes the projection between a single surface point $\hat{\mathbf{x}}$ and the N_δ support points inside the support radius. Other than in the GSB method, in the MLS method the projection matrix is built row by row for each surface point separately. The final projection matrix \mathbf{P} is assembled from the M row matrices of all surface points. For each surface point a $Q \times Q$ -matrix \mathbf{A} has to be set up and inverted. Its condition number and thus its invertability depends on the number of support points and their spatial arrangement. Practical experience has revealed that the regularisation of the linear systems of equations [13] by left multiplication of $[\mathbf{m}(\bar{\mathbf{x}}_n)]$ is highly detrimental to its condition number. A more accurate and robust numerical solution can be achieved if instead for each surface point the N_δ overdetermined systems of equations

$$\Phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) [\mathbf{m}(\bar{\mathbf{x}}_n)]^T [\boldsymbol{\alpha}^*(\bar{\mathbf{x}}_n)] = \Phi(\hat{\mathbf{x}}, \bar{\mathbf{x}}_n) \quad (15)$$

are solved with QR decomposition, yielding N_δ tuples of polynomial coefficients $\boldsymbol{\alpha}^*(\bar{\mathbf{x}}_n)$ for unit deflections $u_\lambda^*(\bar{\mathbf{x}}_n)$ at the individual support points. The final row entry to the projection matrix then is

$$\mathbf{P}(\hat{\mathbf{x}}) = \mathbf{m}^T(\hat{\mathbf{x}}) [\boldsymbol{\alpha}^*(\bar{\mathbf{x}}_n)]. \quad (16)$$

Because the interpolation function (11) has only local support, the projection matrix is sparse, greatly reducing memory requirements in comparison with the GSB method.

Investigations using configurations with multiple components revealed that even with these new projection methods explicitly based on interpolation schemes additional interpolation in intersection regions between assemblies cannot be avoided. If the whole configuration is treated as a single assembly during the projection, the resulting deformed wetted surface is contiguous, but extremely distorted. (With the GSB method, its global polynomial term in the interpolation function can even result in an unfeasible propagation of deformations to assemblies not directly connected, e.g. from the main wing to the empennage.)

For the MLS scheme, an additional interpolation in intersection regions has been implemented that works in a similar fashion as the interpolation of the FIE method (3): For a surface point situated in an intersection region, as depicted on the left of Fig. 4, a row entry to the projection matrix $\mathbf{P}_k(\hat{\mathbf{x}})$ is built for the element groups directly assigned. Further entries are built with the structures of each adjacent surface segment. The resulting K row entries are assigned normalised weights w_k according to the surface point's distance a to the intersection curve. Weighted averaging yields the final entry to the projection matrix for the given surface point:

$$\mathbf{P}(\hat{\mathbf{x}}) = \sum_{k=1}^K w_k(\hat{\mathbf{x}}, a_{\text{limit}}) \mathbf{P}_k(\hat{\mathbf{x}}) \quad \text{with} \quad \sum_{k=1}^K w_k(\hat{\mathbf{x}}, a_{\text{limit}}) = 1. \quad (17)$$

This interpolation algorithm exploits the fact that the MLS method builds the projection matrix one surface point at a time. In the GSB method, the projection matrix is created for all points simultaneously. To realise a comparable interpolation, for each assembly a projection matrix would have to be built which relates the points of its surface segment to all neighbouring structures. Only then the average can be taken with suitable weights assigned to the row entry of each surface point. Because of the high memory requirements of just a single projection matrix resulting from the GSB method, comparable interpolation has not been implemented in the ACM.

3.5 Insertion of Additional Support Points

The solution accuracy and the robustness of the MLS projection algorithm improves with increasing number of support points, but only if these offer sufficient information density in all spatial directions. Moreover, with higher number of support points necessarily also the support radius becomes larger; hence the desirable locality of the projection diminishes. These two problems can be alleviated by using an idea also put forward by Quaranta et al. [23]: It is not the actual nodes of the

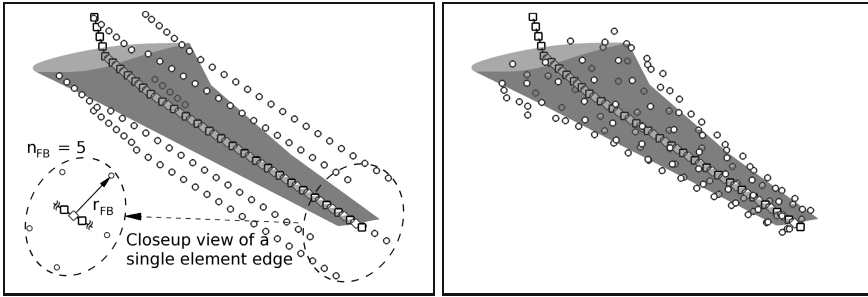


Fig. 5 *left*: Arrangement of the “fishbones” generated for a beam model. The structural nodes are represented by squares, the edge midpoints by diamonds and the additional support points by circles. Here the radius r_{FB} is fixed to a value approximately half the average chord length. The number of additional support points per edge midpoint n_{FB} is set to five, which is the recommended value for beam models. *right*: Arrangement of the additional support points after alignment with the wetted surface.

structural model that are taken as supports for the interpolation. Instead, the midpoints of the beam elements, or respectively the edge midpoints of higher-dimensional elements are used. Surrounding these points, additional supports are inserted circumferentially, as shown in the left image of Fig. 5. Quaranta et al. coined the term “fishbones” for this arrangement. The additional support points not only assure adequate information distribution in all spatial directions, but also allow for the simple projection of rotational deformational components at the structural nodes: The rotations are interpolated to the edge midpoints and result in a translation of the additional support points according to their radius r_{FB} . During the converse projection of loads from the wetted surface to the structure the forces projected to the additional support points are combined at the edge midpoints and corresponding offset moments are introduced. The loads at the edge midpoints are then split between the adjacent structural nodes.

The GSB and MLS projection methods were further enhanced by enabling an automatic alignment of the additional support points with the wetted surface, as shown on the right of Fig. 5. To this end, the information which surface points are in the vicinity of each edge midpoint is required. Therefore, in a preparatory step the mapping of supporting edge midpoints to surface points is inverted. The shape of the surface section normal to a given edge is approximated as an ellipse. The additional support points are then inserted along its circumference with equiangular spacing. Special attention is needed when the wetted surface does not cover the whole circumference, like along the fuselage of a half-model suspended in the symmetry plane.

4 Error Sources in Spatial Coupling

In this section, the investigation of error sources in the spatial coupling by means of model problems shall be detailed. Tracing the steps outlined in Section 2, several potential sources of error were identified and looked into during the project MUNA. Three of these investigations are presented here: The influence of the mesh spacing of the structural and surface mesh is determined as well as the effect of the deformation mapping on the shape of the deformed wetted surface. For all three available spatial coupling methods, parameter studies were carried out to ascertain the importance of the user-defined projection parameters.

4.1 CFD Mesh Spacing and Load Distribution

When discretising the solution domains of the flow problem and of the structural problem before a coupled aeroelastic simulation, one would prefer to choose the grid spacings only considering the requirements of the single-field solvers and the desired solution accuracy. Especially one would like to avoid having to match up the discretisations at the coupling surface, which might not be possible at all when reduced structural models are used. Therefore the question arises as to how the choice of mesh spacings of the structural model and of the subsets of the CFD volume mesh representing the wetted surface influence the projection results. The influence on the load projection is most conveniently analysed by regarding the calculation of the consistent nodal loads on the wetted surface together with their projection on the structure.

4.1.1 One-Dimensional Test Setup

First investigations were carried out with one-dimensional test configurations loosely following an approach laid out by Jaiman et al. [17]: The one-dimensional fluid mesh and the structural mesh are colinear, but feature non-matching discretisations. An analytical pressure distribution is applied to the fluid mesh. The equivalent consistent nodal loads are then projected on the structural nodes. Furthermore, a reference load distribution can be obtained by calculating the nodal loads consistent with the pressure distribution directly at the structural nodes. The reference loads can then be used to calculate a relative error ε_F of the projected loads. In the first three diagrams of Fig. 6, the analytical pressure distribution is plotted over the length of the one-dimensional domain together with the forces acting on the fluid nodes and on the structural nodes. The fluid mesh is discretised with an increasing number of elements n_{CFD} , whereas the number of elements on the structural side is kept constant at $n_{\text{CSD}} = 20$. In this example, linear shape functions are used for both the calculation of the consistent nodal forces and for their projection with the FIE method. With the coarser CFD meshes the distribution of the projected forces on the structural nodes is highly irregular. Parameter studies indicate that the mesh spacings on

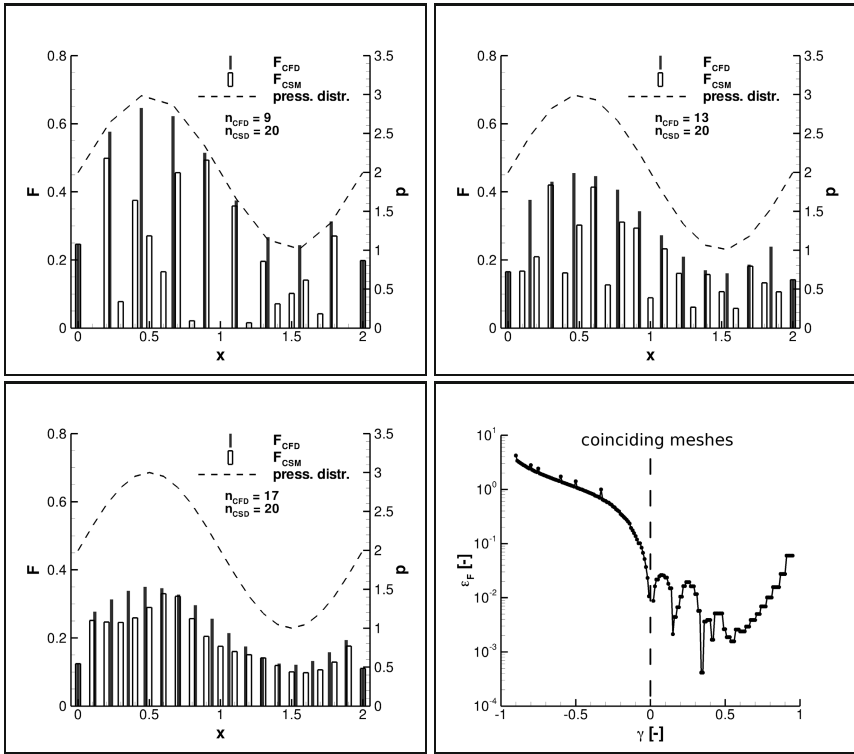


Fig. 6 Influence of the mesh ratio on the load distribution on a one-dimensional test configuration. *top left, top right and bottom left:* Consistent nodal forces on the fluid nodes and projected forces on the structural nodes for fluid meshes with 9, 13 and 17 elements. The number of structural elements is kept at $n_{CSD} = 20$. *bottom right:* Relative error ϵ_F plotted over the mesh ratio γ .

both sides have to be fairly similar in order to achieve a sufficiently regular load distribution. The graph on the bottom right of Fig. 6 underscores this result. It shows the relative error plotted over the mesh ratio $\gamma = \frac{1-n_{CSD}/n_{CFD}}{1+n_{CSD}/n_{CFD}}$. Values from -1 to 0 imply a fluid mesh coarser than the structural mesh, and values between 0 and 1 a finer one. For negative values of γ the error is high, and only approaching $\gamma = 0$ it decreases to an acceptable level. In this example, for $\gamma = 0$ the structural nodes and the fluid nodes are placed at the same coordinates and the relative error becomes zero (not shown in the logarithmic diagram).

4.1.2 Beam Model Test Setup

While these studies revealed first clues of the influence of the structural and CFD surface mesh spacings, the test configuration is very abstract and the conclusions

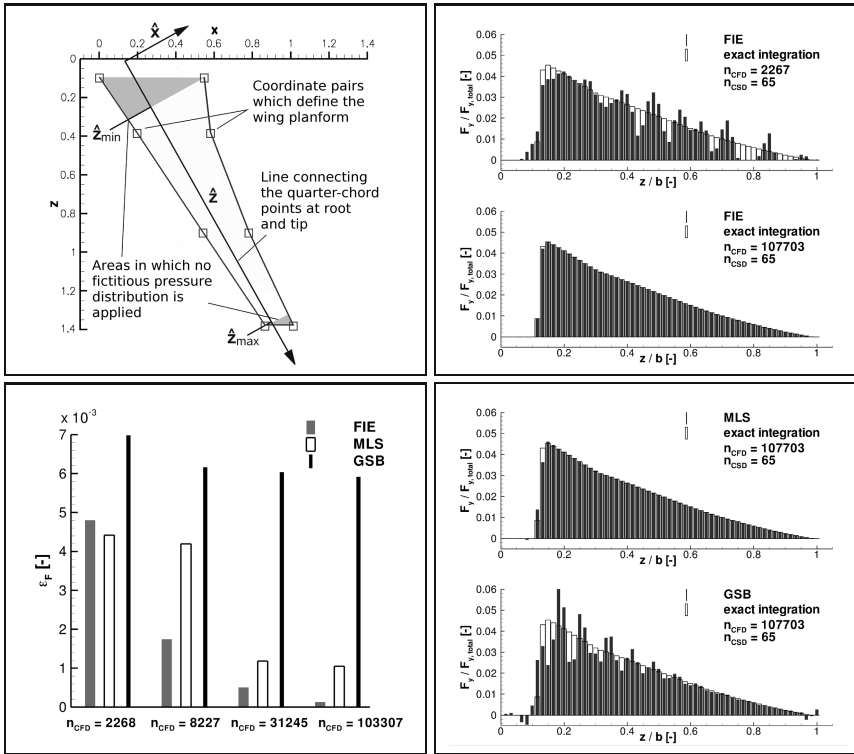


Fig. 7 Influence of the CFD mesh resolution on the structural load distribution. *top left*: Definition of the fictitious pressure field imposed on the wing. *top right*: Nodal forces in flap-wise direction resulting from the pressure distributions on the coarsest and the finest surface mesh and the projection with FIE. These loads are juxtaposed with those obtained by exact integration of the pressure distribution. *bottom left*: RMS error of the nodal forces in flap-wise direction for the three projection methods. *bottom right*: Nodal forces in flap-wise direction resulting from the pressure distributions on the finest surface mesh and projection with MLS and with GSB.

may not be transferable to real-world problems. Further test configurations were created for use together with the stand-alone version of the ACM which are based on the wetted surface of the HIRENASD wing [4]. The resolution of the wetted surface was varied again to quantify its effect on the structural load distribution. Four refinement levels with 2268, 8227, 31245 and 107703 nodes were realised by extracting different multigrid levels from a structured FLOWer mesh. Certainly, with an actual flow solver any change of the mesh alters the flow solution. Because this problem is specific to the single-field solver and not to the spatial coupling method, a fictitious pressure distribution was again employed here. It is defined in terms of the rotated planform coordinates (\hat{x}, \hat{z}) shown on the top left of Fig. 7. The \hat{z} -axis is chosen to connect the quarter-chord points at root and tip. The pressure distribution describes a quarter cosine wave along the \hat{z} -axis and half a sine wave along

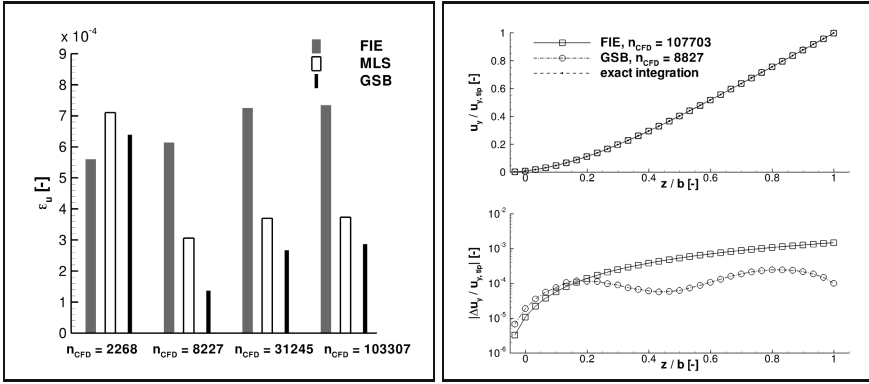


Fig. 8 Influence of the CFD mesh resolution on the structural deformation. *left*: RMS error of the flapwise bending deflections resulting from the different load distributions. *right*: Comparison of the actual bending deflections and the distributed error for the case with the highest deflection error (FIE, $n_{CFD} = 103307$) and the lowest (GSB, $n_{CFD} = 8227$).

the perpendicular coordinate direction \hat{x} . In the shaded areas the fictitious pressure distribution is zero. It is applied with opposite signs to both surfaces of the wing to produce a net positive bending moment in flap-wise direction. From the exact integration of the pressure distribution a line load along the \hat{z} -axis is obtained. If a beam stick model has parallel orientation, the line load can be divided consistently among the structural nodes to obtain a reference load distribution. This serves to define an error incurred by each projection scheme in dependence of the surface mesh resolution. Moreover, with the reference load distribution the structural deformation is computed and compared to those of the projected loads. As a measure for the error the root mean square (RMS) of the nodal forces in flap-wise direction, normalised by the total force in flap-wise direction, is used:

$$\epsilon_F = \sqrt{\frac{1}{n_{CSD}} \sum_{i=1}^{n_{CSD}} \left(\frac{F_{y,i} - F_{y,i,exact}}{F_{y,tot}} \right)^2} \quad \text{with} \quad F_{y,tot} = \sum_{i=1}^{n_{CSD}} F_{y,i,exact} \quad (18)$$

This is plotted in the bottom left image of Fig. 7 and different trends are apparent for the projection schemes. FIE exhibits a strong reduction in the error when the surface mesh resolution is increased, whereas the error with GSB remains almost constant. MLS ranges in between. In contrast to the FIE method the MLS and GSB methods are not able to capture the discontinuous onset of the pressure distribution near the root because they rely on finite support radii. Even with the finest surface mesh the load distribution at the root remains smeared out. Over the remaining span the MLS method approximates the exact loads well, whereas with the GSB method there are still large discrepancies, as can be seen from the bottom right diagrams of Fig. 7. In the left image of Fig. 8 the resulting bending deflections in flap-wise directions are compared by means of the RMS differences of the nodal deflections normalised by the tip deflection

$$\varepsilon_u = \sqrt{\frac{1}{n_{\text{CSD}}} \sum_{i=1}^{n_{\text{CSD}}} \left(\frac{u_{y,i} - u_{y,i,\text{exact}}}{u_{y,\text{tip}}} \right)^2}. \quad (19)$$

It should be noted that because of their different normalisation the error values ε_u and ε_F cannot be directly compared.

For the investigated straight beam model, the differences in the load distribution do not translate in large differences in deformations. This is documented in the right diagram by a comparison of the bending deflections and their deviations from the reference distribution for the cases with the highest and the lowest total deflection error. The deformed structural models are almost identical because in all projection methods the redistribution of the bending forces along the beam axis is compensated by offset moments.

As the structural model is straight and the configuration comprises only a single assembly, no interpolation parameters come into play with the FIE method (cf. Chap. 3.2). For both the MLS and the GSB method, eight edge midpoints were set as supports, with $n_{\text{FB}} = 5$ additional support points each. The radius of the support points was chosen as $r_{\text{FB}} = 0.15$ m, which is approximately half the mean chord length. Quadratic polynomials were used for the global contribution to the interpolation function of the GSB method and for the local interpolation functions of the MLS method.

4.1.3 Shell Model Test Setup

With the beam model test configuration, the different interpolation schemes and surface mesh resolutions do not produce profound local load incidence effects or major differences in the global deformation. A third test setup was investigated which bears more resemblance to a real-world configuration. The wetted surface is the HIRENASD wing scaled to a half-span of 29 m. The structural model is a shell model kindly supplied by the Institute of Aircraft Design and Lightweight Structures (IFL) of the Technical University of Braunschweig. It is akin to a modern transport aircraft wing box and has been dimensioned to real-world design loads. As such it has a realistic ratio of local sheet flexibility to total cantilever flexibility. The model is depicted in the left image of Fig. 9; a detailed description can be found in the contribution of Reich et al. in this volume. Also in this test case on wetted surfaces with varying resolutions a fictitious pressure distribution was applied. However, it cannot be exactly integrated here and thus no reference load or deformation distribution is available. In the right image of Fig. 9, the loads projected on the structure with the MLS method are depicted for the coarsest and the finest surface mesh. With decreasing number of points on the wetted surface, the absolute values of the aerodynamic surface loads increase due to the larger area of each individual surface cell. At least for the structured CFD meshes used here, simultaneously a concentration of the surface loads occurs. Potentially both can cause local load incidence effects, i.e. local “bumps” on the shell model which are then projected back to the wetted surface and might locally alter the flow field. The bumps are a

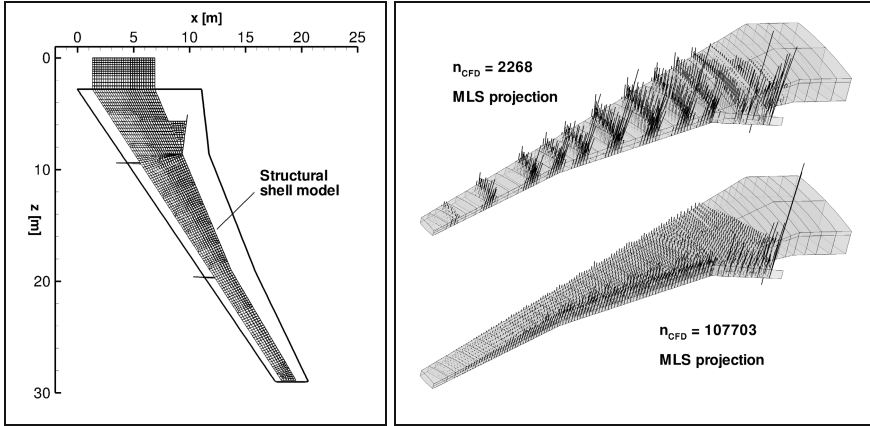


Fig. 9 *left*: Planform of the scaled HIRENASD wing and structural shell model created at the Technical University of Braunschweig. *right*: Comparison of the force vectors resulting from the fictitious pressure distribution on the coarsest and on the finest wetted surface and projection with MLS.

result of the mismatch between the structural and fluid meshes; if forces rather than surface strains are projected, any such appearance must be examined: does it represent a valid structural deformation or is it merely an artifact of the spatial coupling method.

In Fig. 10 the deviations in load and deformation distributions are compared for the FIE and MLS projection and the four investigated surface meshes. No results were obtained with the GSB method owing to the high numerical effort brought about by the inversion of the RBF weights matrix C in Eq. (10). The data plotted in Fig. 10 are not to be understood as absolute projection errors. The reference values do not result from exact solution, which is not available. Here, the results obtained for the surface mesh with 107703 points and the FIE method were chosen, this however does not implicitly make them the “correct” values.

In the top left diagram there is an inherent deviation visible between the load distributions obtained with the projection methods which does not decrease significantly with increasing mesh resolution. The deviations in the deformation distribution, though, are strongly dependant on the mesh resolution rather than on the projection method, as can be seen from the top right diagram. This effect can once again be attributed to the offset moments which partially compensate the differences between the load distributions. A distributed deformational deviation was extracted along a line in span-wise direction and is shown in the bottom left image of Fig. 10. The graphs for the coarsest surface mesh and projection with FIE and MLS are virtually identical. They exhibit the mentioned bumps, but also a global bending deflection higher than in the reference case. The bumps are in the order of tenths of a per cent of the total bending deflection, which for this model equates to local

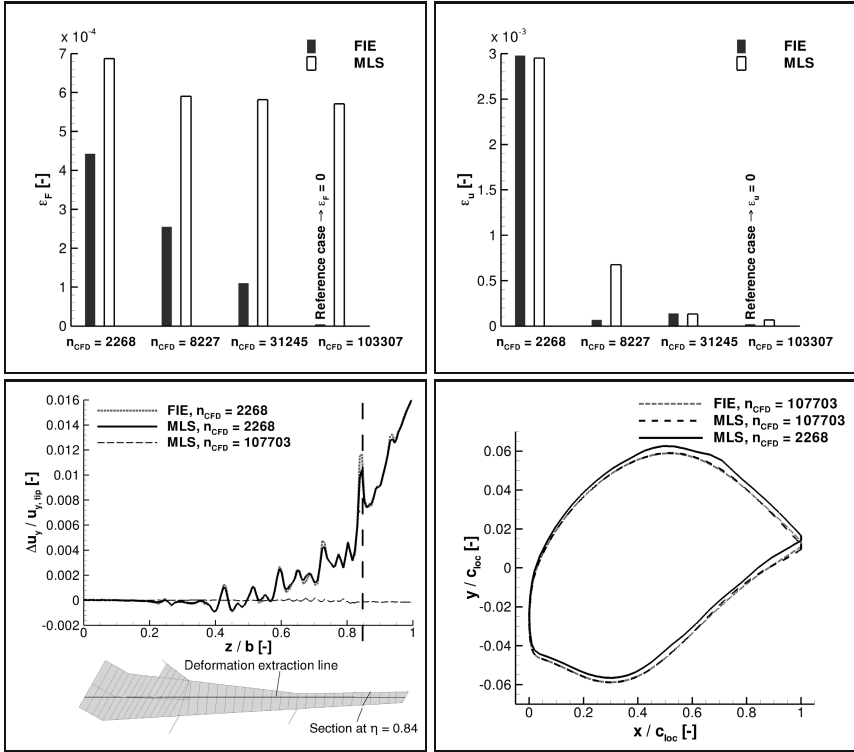


Fig. 10 top left: RMS deviations of the forces in flap-wise bending direction according to Eq. (18). In all plots shown in this panel the finest wetted surface with 107703 nodes and FIE projection provides the reference values. top right: RMS deviations of the flap-wise bending deflections according to Eq. (19). bottom left: Local deviations of the bending deflection in flap-wise direction for the coarsest surface mesh and projection with MLS and FIE. The deflections are extracted along the length of the suction side of the wing box. Additionally the deviations with the finest surface mesh and MLS projection are shown. bottom right: Comparison of sections through the wetted surface at the spanwise position $\eta = 0.84$.

differences in the contour of several millimetres. The difference in global deflection is in the order of one per cent. The deviations between the projection methods for the finest mesh level are also plotted. They are close to zero all along the span. For a rough assessment of the influence of the differing structural deformations on the shape of the wetted surface, in the bottom right image two sections through the wetted surface at the spanwise position $\eta = 0.84$ are superimposed, which is where the distinct peak in the deviations is visible in the bottom left image. This comparison is slightly marred by the fact that the respective projection methods are applied twice, first for the loads and again for the deflections. Nonetheless, both the difference in global deflection and a bump on the suction side are apparent for the coarsest wetted surface. It can be concluded that with thin-walled structural shell models and coarse

CFD meshes load incidence effects can indeed have an influence on results, but it is seen to diminish rapidly with the finer CFD meshes. Even though such coarse meshes are not regularly used for standard steady simulations, they still play a role in unsteady simulations and in design, where accuracy is sacrificed for the sake of solution speed. For instance during the preceding project MEGADESIGN [20], a design case was investigated using a volume CFD mesh with approximately 170000 points; 4425 points thereof made up the surface mesh.

4.2 Influence of Projection Parameters

The spatial coupling methods available in the ACM all base the transfer of loads and deformations on geometric neighbourhood relations between the wetted surface and the structure. For this, the methods require different additional interpolation parameters. As explained in Chap. 3.2 the FIE method for beam elements has three additional interpolation parameters: The weighting parameters β_{limit} and a_{limit} apply for projections of surface points in the vicinity of kinks of the beam model. The width of the intersection region a_{limit} has to be defined if assemblies border on each other. For shell and volume models, the FIE method currently does not call on interpolation parameters. Once it has been extended to configurations with multiple assemblies, a_{limit} also will come into play. The GSB and the MLS methods are derived from general interpolation algorithms. They require the definition of the (minimum) number of supports N_δ or N_M , the number of additional support points per edge midpoint n_{FB} , their radius r_{FB} , the polynomial degree of the interpolation functions (7) or (11) and the type of RBF function ϕ . The MLS method is also suitable for configurations with multiple assemblies and thus again a_{limit} has to be provided. None of these parameters is directly based on physical considerations, so the optimal values are not obvious.

All interpolation parameters of the FIE method required for beam models shall be examined here, whereas for the MLS and GSB methods only a selection is presented in detail. As became clear from the results presented in Chap. 4.1.2, distinct load distributions can result in identical deformation distributions. Therefore, the influence of the projection parameters is evaluated via the deformation projection. In order to have identical input data for all cases, no aerodynamic loads are imposed on the wetted surface. Instead, a force distribution is applied directly to the structural nodes. The structural deformations then become independent of the projection method. The differences in the shape of the wetted surface can consequently be attributed solely to the deformation projection. Also in this investigation exact reference solution is not available, and as in the previous chapter the deviations should not be interpreted as absolute errors.

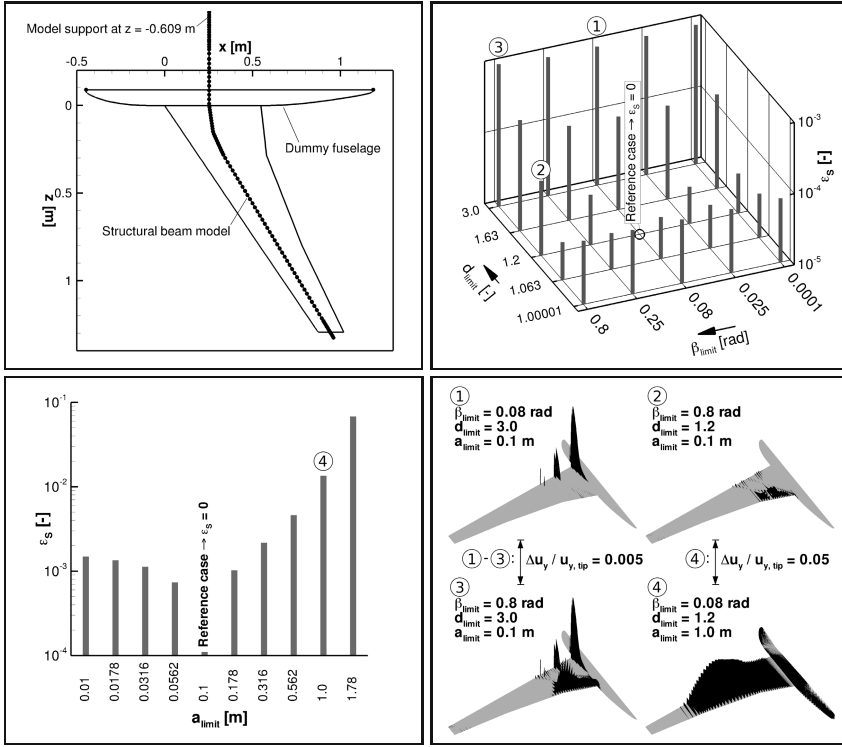


Fig. 11 *top left*: Planform of the HIRENASD wing and its structural beam model. The model support is outside the pane at $z = -0.609$ m. The dummy fuselage is not physically connected to the wing. It is also represented as a component in the beam model, but its nodes are all clamped. *top right*: RMS deviations of the wetted surface deflections relative to those obtained with the default settings. The parameters β_{limit} and d_{limit} are varied and a_{limit} is kept constant at 0.1 m. *bottom left*: RMS deviations of the wetted surface deflections relative to those obtained with the default settings. a_{limit} is varied, while the remaining parameters are kept constant at $\beta_{limit} = 0.08$ rad and $d_{limit} = 1.2$. *bottom right*: Distributed values of the normalised deflection difference $|\Delta u_y / u_{y,tip}|$ for the four parameter combinations marked in the top right and bottom left diagrams.

4.2.1 Interpolation Parameters of the FIE Method for Beam Models

The interpolation parameters of the FIE method are the angular limit of the interpolation area in the vicinity of beam kinks β_{limit} , the limit ratio of projection distances in the vicinity of beam kinks d_{limit} , and the width of the intersection region between assemblies a_{limit} . They are varied and their respective influence on the shape of the deformed wetted surface is assessed. The test case is the HIRENASD configuration with a dummy fuselage, which constitutes a second assembly next to the wing. The CFD surface mesh sketched in the top left of Fig. 11 has 46919 points and the structural beam model has 654 nodes. These components are subjected to forces and moments which resulted from a previous aeroelastic simulation. These are kept

constant during the parameter study. The deformed wetted surface acting as a reference for the deviations was obtained with the default settings of the ACM, which are $\beta_{\text{limit}} = 0.08 \text{ rad}$, $d_{\text{limit}} = 1.2$ and $a_{\text{limit}} = 0.1 \text{ m}$. This last value amounts to approximately 8% of the model half-span of 1.29 m. As the parameters d_{limit} and β_{limit} both relate to the interpolation in the vicinity of kinks, these parameters are studied together. The RMS deviation of the surface point coordinates ε_S is determined by analogy to ε_u as defined in Eq. (19). Also in this case only the differences in the flap-wise deflection are considered.

In the top right diagram of Fig. 11 the RMS deviations of the normalised deflection are plotted over the investigated parameter combinations. Whereas the minimum valid values of β_{limit} and d_{limit} are defined inside the ACM, the upper values were chosen arbitrarily. The choice of d_{limit} has a more profound effect on the shape of the wetted surface than β_{limit} . There is a weak interdependence visible between the two parameters. The indication of a global RMS value is somewhat misleading here, as both parameters lead to highly localised deviations. This can be seen in the bottom right image. The distributed values are displayed as bars over the configuration's planform for combinations of the maximum and the default settings of β_{limit} and d_{limit} . Distinct peaks close to the leading edge are apparent for the maximum value of d_{limit} , whereas for the maximum value of β_{limit} there are differences visible in the wedge-shaped areas of non-unique projection mentioned in Chap. 3.2. The choice of the width of the intersection region between fuselage and wing not only has a more widespread influence. It also produces deviations an order of magnitude higher and it thus gives far higher RMS errors. To put the given normalised distributed values into perspective: Assuming a bending deflection of 5% of the model half-span of 1.29 m, the peak deviation for parameter combination ③ is less than 0.5 mm, but for setting ④ with $a_{\text{limit}} = 1.0 \text{ m}$ it exceeds 3 mm.

4.2.2 Interpolation Parameters of the MLS and GSB Methods

The newly-implemented projection methods MLS and GSB have more control parameters than the FIE method. Only the (minimum) number of supporting edge mid-points N_M , the polynomial order of the interpolation functions and the radius of the additional support points r_{FB} are examined here. The results for the remaining parameters are briefly summarised beforehand. As reference for the deviations the deformed wetted surface obtained by application of the FIE projection with default parameter settings is used.

The number of additional support points to be generated around each edge midpoint n_{FB} depends on the type of structural model. In previous extensive tests, $n_{\text{FB}} = 5$ was determined as the recommended number for beam models. With lower numbers reliable solutions could neither be obtained with the MLS method nor with the GSB method. Then again, greater numbers do not yield noticeable improvements in robustness or accuracy. With structural models consisting of higher-dimensional elements, the number of additional support points generally can be reduced. For shell models $n_{\text{FB}} = 2$ is often sufficient. For volume models, additional support

points are likely to be omitted. The choice of the RBF has only very small influence on the solution both in the MLS and the GSB method. Wendland RBFs with different orders of smoothness have been tried out as well as other RBF with compact support, like Euclid's hat functions or the Thin Plate Spline, with hardly any effect on the wetted surface. Because the mechanism to perform the interpolation in intersection regions between assemblies in the MLS method is very similar to the mechanism in the FIE method, the same effects on the wetted surface can be expected from variations of a_{limit} .

The configuration used here is similar to the one presented in the previous section in Fig. 111, but without the dummy fuselage. It comprises only one component and thus allows direct comparison between GSB and MLS. The surface mesh has 31245 points. The structural model with 654 nodes and the applied load distribution are the same as before.

4.2.3 Interpolation Parameters of the GSB Method

In the top left image of Fig. 112, the RMS deviations in the flap-wise bending deflection are plotted over the minimum number of supporting edge midpoints $N_{M,\text{min}}$. The radius of the additional support points was fixed at $r_{\text{FB}} = 0.15$ m. In the GSB method the actual number of support points differs all over the wetted surface, as was explained on page 193. The support points of a given surface point are all the edge midpoints and additional support points within the support radius δ . The deformed wetted surface used here as a reference was obtained with the FIE method and its default parameter settings. The deviations do not seem to be influenced by the number of support points. The RMS values hover at 1.7% for linear global interpolation functions and at 0.12% for quadratic ones. Yet the wetted surfaces obtained with MLS are not completely identical, as is documented in the bottom left diagram. It shows the RMS deviation relative to the deformed surface resulting from the GSB method with $N_{M,\text{min}} = 8$. These deviations, however, are at least one order of magnitude smaller than those relative to the FIE reference case.

The top right image of Fig. 112 shows the influence of the radius of the additional support points. Eight fixed radii are investigated as well as the alignment of the additional support points with the wetted surface. The number of supporting edge midpoints is $N_{M,\text{min}} = 8$; the reference wetted surface as before resulted from the FIE method. The better the spatial arrangement of the additional support points approximates the wetted surface, the lower the average deviations come to be. For both linear and quadratic global contributions to the interpolation function the smallest RMS values are achieved with an alignment of the additional support points. Next best is $r_{\text{FB}} = 0.15$ m, which is approximately half the mean chord length. Between $r_{\text{FB}} = 0.276$ m and $r_{\text{FB}} = 0.474$ m a marked increase in deviations occurs. This is because r_{FB} becomes larger than the fixed support radius δ , and the additional support points cease to come into play. Case ④ in the bottom right image shows the distribution of the normalised deviations. These increase over the length of the span and values of $|\Delta u_y / u_{y,\text{tip}}| > 2$ are reached at the tip. Cases ① and ② highlight a general

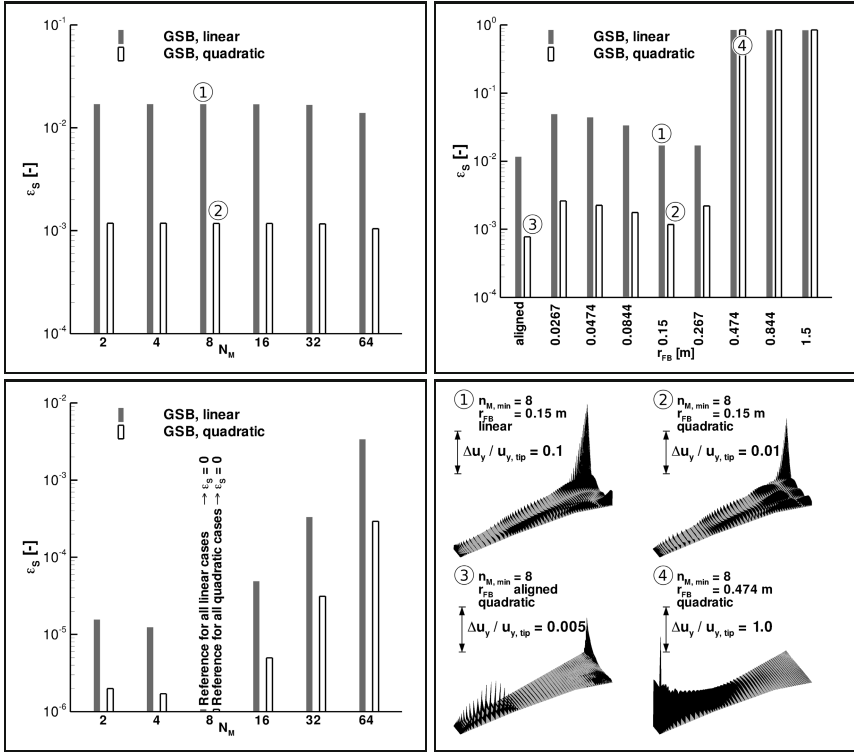


Fig. 12 RMS deviations of the wetted surface deflections of the GSB method relative to those obtained with the FIE method and its default parameter settings for the projection parameters. *top left*: Variation of the minimum number of supporting edge midpoints between $N_M = 2$ and $N_M = 64$ with $r_{FB} = 0.15$ m. *top right*: Variation of the radius of the additional support points between $r_{FB} = 0.0267$ m and $r_{FB} = 1.5$ m with $N_M = 8$. *bottom left*: Variation of the minimum number of supporting edge midpoints between $N_M = 2$ and $N_M = 64$. In this diagram the deflected surface obtained with GSB and $N_M = 8$ is the reference for the deviations. These are determined separately for linear and quadratic global contributions to the interpolation function. *bottom right*: Distributed values of the normalised deflection difference $|\Delta u_y / u_{y, \text{tip}}|$ for the four parameter combinations marked in the top diagrams. Note the different scales for each case.

problem of the GSB method: The deformation distribution is generally not approximated well by the global polynomial contribution to the interpolation function, not even by a quadratic one. This is to be compensated by the local RBF contributions of the edge midpoints and of the additional support points. The deviations become large in regions of the wetted surface far-off the support points, for instance at the leading and trailing edges. Consequently, case ③ with alignment of the additional support points with the wetted surface reveals significant improvement over cases with fixed values of r_{FB} .

Ahrem et al. [11] propose breaking down the configuration into sections and applying the GSB method on each one by itself. The projection results are smoothly interpolated by a partition of unity algorithm. In each section a different fit for the global polynomial contribution is obtained which results in a better approximation of the wetted surface. Simultaneously the memory requirements and the numerical effort are reduced. The downside is the introduction of yet another interpolation scheme, and this approach has not been included in the ACM.

4.2.4 Interpolation Parameters of the MLS Method

The parameter variations presented for the GSB method were repeated with the MLS method. In the top left image of Fig. 13 the RMS deviations are shown in dependence of the number of support midpoints N_M . As before the reference deformed wetted surface is provided by the FIE method with default parameter settings. In the MLS method, during selection of support points an edge midpoint is always considered jointly with its surrounding additional support points. Hence the total number of support points of a given surface point is always a fixed multiple of the supporting edge midpoints $N_S = (n_{FB} + 1) \times N_M$, at least outside of intersection regions. For two edge midpoints the MLS projection with quadratic interpolation functions fails; the resulting deformed wetted surface is completely distorted. Even with the numerically more robust formulation (15) the MLS method requires a minimum number of support points which should be at least twice the number of monomials Q . With four or more supporting edge midpoints the MLS method results in only small deviations relative to the FIE method. Neither with linear nor with quadratic interpolation functions does the deviation exceed 10^{-4} . In the inset of the top left diagram, a typical distribution of the normalised deflection differences $|\Delta u_y / u_{y,tip}|$ is given. These are present all over the wing. The finite support radius of the MLS method slightly smears the deformation during projection as compared to the FIE method's. This represents a systematic discrepancy largely independent of the choice of MLS parameters. There are minor differences between the resulting wetted surface, though, as can be seen from the top right diagram. Here, the deformed wetted surface obtained with MLS for $N_M = 8$ and $r_{FB} = 0.15$ m is the reference. The different approximation orders of the interpolation functions are weakly reflected in the RMS values, as is the number of support points.

In the bottom images of Fig. 13 the influence of the radii of the additional support points is examined. The bottom left diagram underscores the systematic discrepancy between FIE and MLS which goes completely unaffected by the choice of r_{FB} . Again the smeared projection of deformations with MLS is apparent in the inset. The largest deviations occur in the region of the beam kink, where the shape of the surface deformed with FIE also depends on the choice of projection parameters (see cases ① to ③ in Fig. 11). In the bottom right diagram of Fig. 13, the deformed wetted surface obtained with MLS and $N_M = 8$ and $r_{FB} = 0.15$ m is again the reference. For all contemplated parameter settings the deviations are very small. The systematic discrepancy between the results for linear and quadratic interpolation functions is due to the single peak at the leading edge wing root visible in the

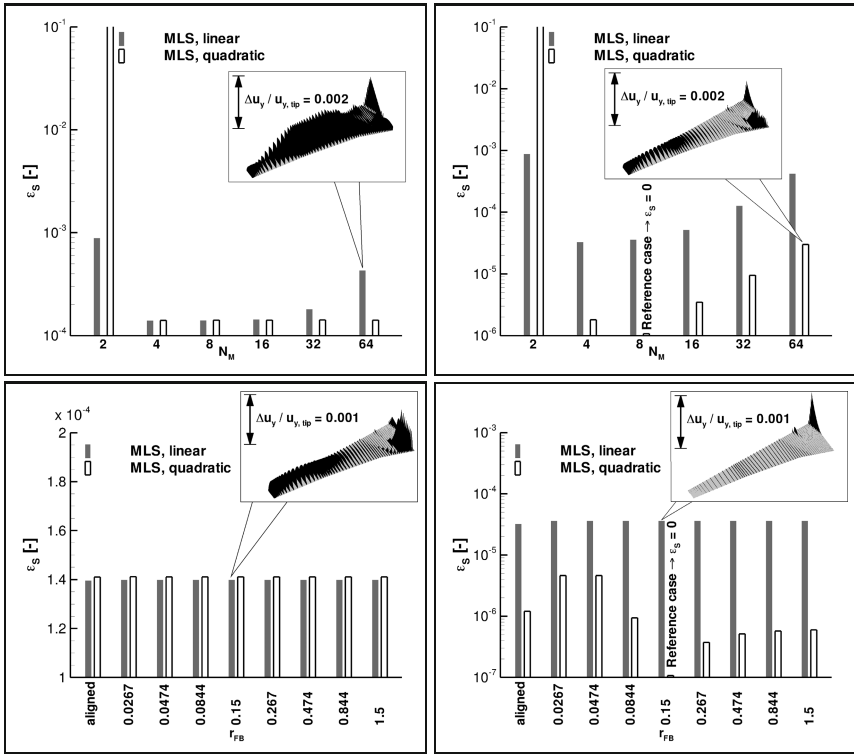


Fig. 13 RMS deviations of the wetted surface deflections of the MLS method. *top left*: Variation of the minimum number of supporting edge midpoints between $N_M = 2$ and $N_M = 64$ with $r_{FB} = 0.15$ m. The deflected surface obtained with FIE and default parameter settings is the reference for the deviations. *top right*: Variation of the minimum number of supporting edge midpoints between $N_M = 2$ and $N_M = 64$. The deflected surface obtained with MLS and $N_M = 8$ and $r_{FB} = 0.15$ m is the reference for the deviations. *bottom left*: Variation of the radius of the additional support points between $r_{FB} = 0.0267$ m and $r_{FB} = 1.5$ m with $N_M = 8$. The deflected surface obtained with FIE and default parameter settings is the reference for the deviations. *bottom right*: Variation of the radius of the additional support points between $r_{FB} = 0.0267$ m and $r_{FB} = 1.5$ m with $N_M = 8$. The deflected surface obtained with MLS and $N_M = 8$ and $r_{FB} = 0.15$ m is the reference for the deviations.

inset of the bottom right image. This peak can be traced to the lack of additional support points in its vicinity, as can be identified in Fig. 5.

4.2.5 Comparison of the Projection Methods

The projection methods FIE, GSB and MLS were applied to similar configurations and at least a tentative comparison of their relative merits is possible. The projection mechanism of the FIE method is completely in line with beam theory in regions

Table 1 Memory consumption and execution time of the ACM with the available projection methods. Runs were carried out on a single 3.0 GHz Intel Xeon processor. The memory requirements include the storage of the system matrices and the solver workspace.

Configuration	HIRENASD wing with 31245 surface points, beam model with 653 structural elements			Scaled HIRENASD wing with 31245 surface points, shell model with 9352 structural elements	
Projection method	FIE	MLS	GSB	FIE	MLS
Duration of first coupling iteration [s]	3.40	3.47	258.84	204.88	86.29
Duration of subsequent coupling iterations [s]	0.06	0.05	1.45	0.86	0.80
Peak total memory requirement [MByte]	38.8	135.0	1617.6	1727.1	1740.3

where the beam axis is straight and does not have intersections. For the investigated test configuration, this region is the outboard part of the wing. In the vicinity of the beam kink the choice of the projection parameters β_{limit} and d_{limit} has a moderate influence on the local shape of the deformed wetted surface. Correct or optimal values for either are hard to ascertain, and the default values are the results of experience. Approximate values for the width of the intersection region a_{limit} can be determined by common sense, but the choice has a more profound effect on the shape of the wetted surface. While not shown here, fundamentally the same behaviour regarding a_{limit} can be expected for the MLS method. This method has the advantage that the number of support points n_{δ} and the radius of the additional support points r_{FB} can be varied over a large range with only marginal effects on the shape of the deformed wetted surface, independently of the polynomial order of the interpolation function. Compared to the FIE results the deformation distributions are slightly smeared during projection. Further differences to the FIE results are visible in the vicinity of kinks where the FIE method does not present a valid absolute reference either. Finally, the GSB method exhibits the strongest dependency on the choice of projection parameters. A high polynomial order of the global contribution to the interpolation function is beneficial, as it should already represent the deformation field as well as possible. The local RBF contributions have to make up for the difference between the supplied deformation distribution and its global approximation, so that the placement and number of supports bear a special importance.

From the user perspective, any projection method should not only be robust and deliver accurate results, but also have low computational resource requirements. The FIE method needs the least memory because in the implementation used in the ACM it does not store the projection matrix explicitly. The MLS method does so, but the resulting matrix is sparse. Discounting additional entries due to interpolation in intersection regions, the number of non-zero entries is $n_{\text{CFD}} \times N_{\delta} = n_{\text{CFD}} \times N_M \times (n_{\text{FB}} + 1)$. With the GSB method though, the projection matrix is dense

and the number of non-zero entries is the product of n_{CFD} and the total number of support points. It is also the method associated with the greatest numerical effort. In Table 1, typical run-times and peak total memory requirements are summarised for the test cases treated in Chapters 4.1.3 and 4.2. With the shell model, the memory requirements are dominated by the structural system and preconditioning matrices. The overhead for the explicit storage of the projection matrix of the MLS method is not significant, as opposed to the beam model test case.

5 Coupled Simulations

In order to assess the influence of the projection method on the actual coupled solution, steady aeroelastic simulations were performed with the ACM in conjunction with the flow solvers FLOWer and TAU. Two different configurations were used. The scaled HIRENASD wing with the structural shell model depicted in Fig. 9 is representative for a real-world transport aircraft wing. The flow conditions were chosen according to a test case defined in the project MEGADESIGN [20] for the optimisation of an aircraft cruise configuration. The flight speed is $Ma = 0.82$ at standard atmospheric conditions in 11 km altitude. These amount to a Reynolds number of $Re = 43.2 \times 10^6$ and a loading factor of $q/E = 0.144 \times 10^{-6}$. This value is the ratio of the free-stream dynamic pressure and Young's modulus of the structure and is a dimensionless number that characterises aeroelastic coupling effects. The second test configuration employed is the HIRENASD wing in its original size and the beam structural model displayed in Fig. 11 representing the structure of the actual wind tunnel model. Results are shown here for the conditions of the experimental polar 250 [4]: $Ma = 0.80$, $Re = 23.5 \times 10^6$ and $q/E = 0.48 \times 10^{-6}$. In both test cases, the dummy fuselage was not taken into consideration. Identical CFD meshes were scaled to match both configurations. The block-structured mesh for FLOWer is the volume mesh from which the already-presented wetted surface with 31245 points was extracted and has about 2.8 million points. The hybrid-unstructured mesh for TAU has 12.6 million points and a wetted surface with 188983 points. Only results obtained with the FIE and MLS methods are compared.

In Fig. 14 the results for the transport aircraft wing configuration are presented. With the MLS method, quadratic interpolation functions were used with $n_{\text{FB}} = 2$ and $N_M = 67$. In the top diagrams, the lift coefficients and the flap-wise bending deflections at the wing tip are plotted over the angle of attack. For both magnitudes the same observation can be made: The projection methods deliver very similar distributions with the same flow solver, but between the results of FLOWer and TAU there is a difference of around 15%. In the bottom left diagram the relative errors in lift and deflection are given. The influence of the projection method is singled out; the deviations between the flow solvers are not shown. The relative error in the tip deflection $|\Delta u_{y,\text{tip}}/u_{y,\text{tip}}|$ is greater than the values obtained in the previous tests (see Fig. 12). During actual coupled simulations the differences between the projection methods affect the structural load distribution as well as the shape of the deformed

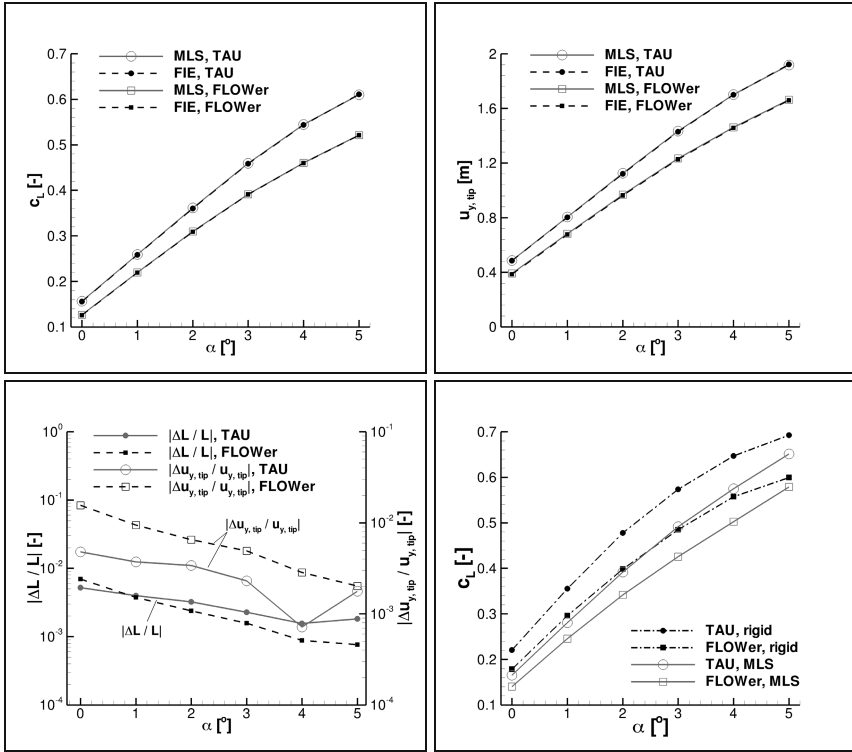


Fig. 14 Results of aeroelastic simulations for the scaled HIRENASD wing and the IFL structural shell model at $Ma = 0.82$, $Re = 43.2 \times 10^6$, $q/E = 0.144 \times 10^{-6}$ and $0^\circ \leq \alpha \leq 5^\circ$. Computations were carried out with FLOWer and TAU coupled with the ACM. FIE and MLS were used as projection methods. In all examples the LEA $k - \omega$ turbulence model and a central differences discretisation in space were applied. *top left:* Lift polars. *top right:* Flap-wise tip bending deflections. *bottom left:* Relative errors in lift coefficients and in bending deflections between the projection methods. The comparisons are only carried out between the distributions obtained with the same flow solver. *bottom right:* Lift polars disregarding aeroelastic deformation superimposed on those obtained with MLS. In a departure from the previously shown results, these were obtained with the scaled HIRENASD wing with dummy fuselage.

wetted surface and then feed back to the aerodynamic load distribution of the next coupling iteration. The deviations are still inside engineering limits, though, ranging between one per cent and one per mil. In the bottom right diagram, the lift distributions obtained by “pure” CFD disregarding aeroelastic deformations are additionally provided. Already here the differences are apparent between the flow solvers (These results were obtained for the scaled HIRENASD wing with dummy fuselage, hence the lift coefficients considering aeroelastic deformation are slightly higher than in the top left diagram). In Fig. 15, local pressure distributions at the spanwise

positions $\eta = 0.55$ and $\eta = 0.83$ are picked out for an angle of attack of $\alpha = 3^\circ$. In the undeformed configuration TAU predicts a shock position further downstream than FLOWer by about 5% of the local chord length. The higher structural bending moment must then lead to an aeroelastic equilibrium configuration with larger deformations. The differences between the flow solvers are thus aggravated by the aeroelastic coupling. This test case is entirely generic and there are no experimental data that would allow an evaluation of these differences.

An extensive experimental data base is available for the HIRENASD configuration. Here the same tendencies as with the previous configuration are to be observed, as can be gleaned from Fig. 16. The deformation distributions and the lift polars also coincide well for the projection methods, but there are significant differences between the results of the flow solvers. Measured lift coefficients are superimposed with the simulation results, but do not give a clear direction. Fig. 17 contains calculated and measured pressure distributions for the two highest angles of attack $\alpha = 2^\circ$ and $\alpha = 3^\circ$ at the spanwise stations $\eta = 0.32$, $\eta = 0.59$ and $\eta = 0.80$. The local pressure coefficients by the MLS method and by the FIE method with identical flow solvers coincide excellently, but significant differences between the flow solvers persist. The higher global lift coefficients achieved with TAU are reflected in the higher suction plateaus and shock positions further downstream. For the current test case, predictions with TAU appear in better agreement with the experiments than the FLOWer results, at least at the inboard sections. At $\eta = 0.80$ neither solver correctly captures the gradient at the downstream end of the plateau and its height is overestimated using TAU. Admittedly, the presented simulations do not capture the aerodynamic influence of the dummy fuselage on the flow field about the wing.

The differences between the results of FLOWer and TAU are presumably caused by the flow meshes or the numerical parameters used at CATS, and are not related to

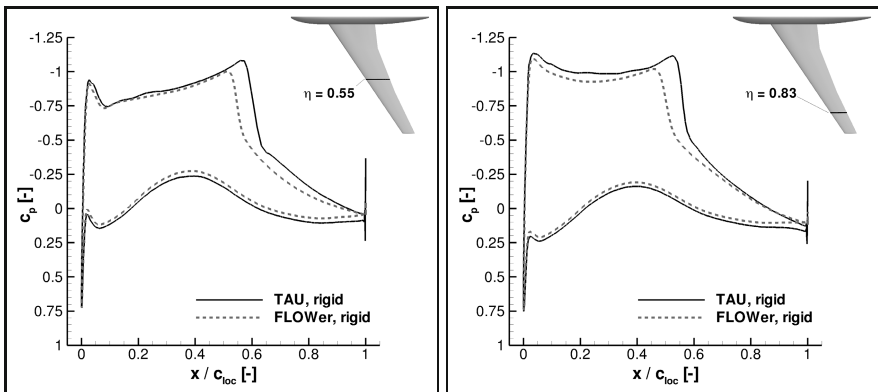


Fig. 15 Pressure distributions obtained with FLOWer and TAU without consideration of aeroelastic deformation at two spanwise stations of the scaled HIRENASD wing with dummy fuselage. Inflow conditions are $Ma = 0.82$, $Re = 43.2 \times 10^6$ and $\alpha = 3^\circ$.

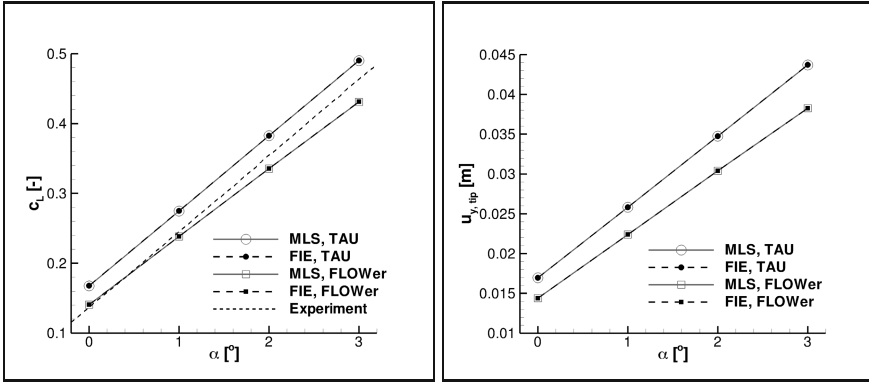


Fig. 16 Results of aeroelastic simulations for the HIRENASD wing with a structural beam model at $Ma = 0.80$, $Re = 23.5 \times 10^6$, $q/E = 0.48 \times 10^{-6}$ and $0^\circ \leq \alpha \leq 3^\circ$. *left*: Lift polars. *right*: Flap-wise tip bending deflections.

the aeroelastic coupling. Further efforts to narrow down the cause of the differences remain to be made, but are outside the scope of the current paper.

6 Conclusion

In this paper, the spatial coupling methods for reduced structural models in the ACM were examined. The existing method for the projection of loads and deformations between the wetted surface and the structure based on FIE was outlined together with the interpolation schemes additionally required for beam models. The alternative methods MLS and GSB and their implementation in the ACM were explained in detail. For the individual steps of the spatial coupling procedure potential error sources were identified. Several test configurations of increasing complexity were used to investigate the importance of the mesh resolutions of the wetted surface and of the structure on the structural load distribution and the resulting deformation field. Only for thin-walled structural models, a significant influence was detected. It manifests itself as local “bumps” on the structure and, after deformation projection, also on the wetted surface. The projection parameters of the FIE, MLS and GSB methods were investigated; their effect on the shape of the deformed wetted surface was determined with two variants of the HIRENASD configuration with and without dummy fuselage and a structural beam model. With the FIE method, only localised effects were apparent in the vicinity of kinks of the beam axis and intersections between assemblies. Here, additional interpolation schemes assure a smooth and contiguous deformed wetted surface. Otherwise, FIE does not rely on interpolation parameters. With MLS and GSB, interpolations are not acting locally, but the projection as a whole is cast as a spatial interpolation problem. (Yet in practice, local interpolations cannot be foregone completely, as without projection results are

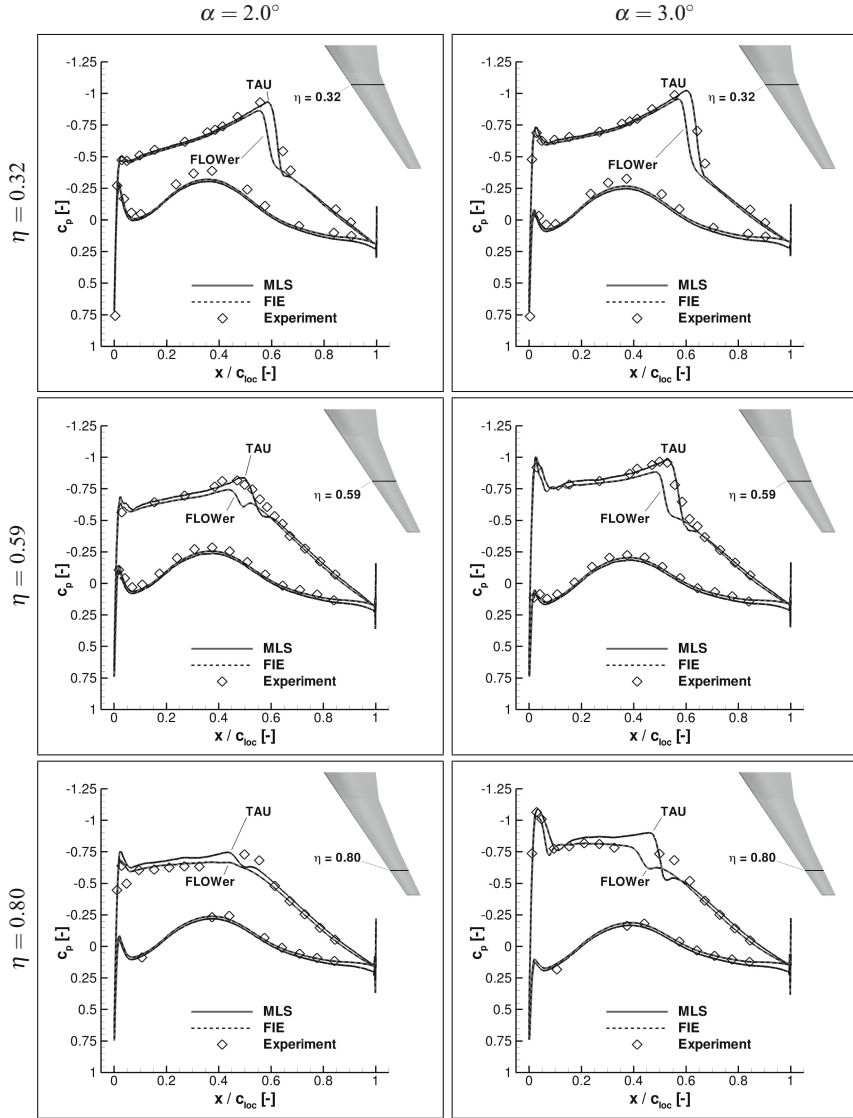


Fig. 17 Measured and computed chordwise pressure distributions for the HIRENASD wing with a structural beam model at $Ma = 0.80$, $Re = 23.5 \times 10^6$ and $q/E = 0.48 \times 10^{-6}$.

unsatisfactory in intersection regions between assemblies.) The MLS method exhibited a low overall dependency on the choice of projection parameters. With the GSB method, the shape of the deformed wetted surface was significantly influenced by the position and number of interpolation support points. By means of coupled simulations with the ACM, and either FLOWer or TAU as flow solvers, the practical

applicability of the MLS method was shown. The differences between the lift polars, deformation distributions and local pressure distributions obtained with MLS and with FIE and identical flow solvers were small. However, in the results achieved with identical projection methods but different flow solvers, significant differences were apparent. These were linked to the flow solvers, respective the input data used, and not to the aeroelastic coupling procedure. Ongoing work on the spatial coupling in the ACM concerns the extension of the FIE method for volume and shell elements to configurations comprising multiple assemblies.

Acknowledgements. Computing resources were provided by the RWTH Aachen University Center for Computing and Communication and supported by the German Research Foundation under GSC 111 (AICES).

References

- [1] Ahrem, R., Beckert, A., Wendland, H.: A New Multivariate Interpolation Method for Large-Scale Spatial Coupling Problems in Aeroelasticity. In: *Int. Forum Aeroel. Struct. Dyn. (IFASD) 2005*, paper IF-073, Munich, Germany (2005)
- [2] Badcock, K.J., Rampurawala, A.M., Richards, B.E.: Intergrid Transformation for Aircraft Aeroelastic Simulation. In: *21st Appl. Aerodyn. Conf., AIAA paper 2003-3512*, Orlando, USA (2003)
- [3] Ballmann, J. (ed.): *Flow Modulation and Fluid-Structure Interaction at Airplane Wings—Research Results of the Collaborative Research Center SFB 401 at the RWTH Aachen University*, Aachen, Germany. Springer (2003)
- [4] Ballmann, J., Boucke, A., Dickopp, C., Reimer, L.: Results of Dynamic Experiments in the HIRENASD Project and Analysis of Observed Unsteady Processes. In: *Int. Forum Aeroel. Struct. Dyn. (IFASD) 2009*, paper IFASD-2009-103, Seattle, USA (2009)
- [5] Beckert, A.: Coupling Fluid (CFD) and structural (FE) Models using Finite Interpolation Elements. *Aerosp. Sci. Technol.* 4, 13–22 (2000)
- [6] Beckert, A., Wendland, H.: Multivariate Interpolation for Fluid-Structure Interaction Problems using Radial Basis Functions. *Aerosp. Sci. Technol.* 5, 125–134 (2001)
- [7] Bendiksen, O.O.: Effect of Wing Deformations and Sweep on Transonic Limit Cycle Flutter of Flexible Wings. In: *Int. Forum Aeroel. Struct. Dyn. (IFASD) 2007*, paper IF-031, Stockholm, Sweden (2007)
- [8] Boucke, A.: *Kopplungswerkzeuge für aeroelastische Simulationen*. Doctoral thesis, RWTH Aachen University (2003)
- [9] Braun, C.: *Ein modulares Verfahren für die numerische aeroelastische Analyse von Luftfahrzeugen*. Doctoral thesis, RWTH Aachen University (2007)
- [10] Cebal, J.R., Löhner, R.: Conservative Load Projection and Tracking for Fluid-Structure Problems. *AIAA J.* 35(4) (1997)
- [11] Chen, P.C., Jadic, I.: Interfacing of Fluid and Structural Models via Innovative Structural Boundary Element Method. *AIAA J.* 36(2), 282–287 (1998)
- [12] Farhat, C., Lesoinne, M., LeTallec, P.: Load and Motion Transfer Algorithms for Fluid/Structure Interaction Problems with Non-Matching Discrete Interfaces: Momentum and Energy Conservation, Optimal Discretization and Application to Aeroelasticity. *Comput. Methods Appl. Mech. Eng.* 157, 95–114 (1998)

- [13] Goura, G.S., Badcock, K.J., Woodgate, M.A., Richards, B.E.: Transformation Methods for the Time Marching Analysis of Flutter. In: 19th AIAA Appl. Aerodyn. Conf., AIAA paper 2001-2457, Anaheim, USA (2001)
- [14] Harder, R.L., Desmarais, R.N.: Interpolation Using Surface Splines. *J. Aircr.* 9, 189–191 (1972)
- [15] Heinrich, R., Kroll, N., Neumann, J., Nagel, B.: Fluid-Structure Coupling for Aerodynamic Analysis and Design — A DLR Perspective. In: 46th AIAA Aerosp. Sci. Meet. Exhib., paper AIAA 2008-561, Reno, USA (2008)
- [16] Hesse, M.: Entwicklung eines automatischen Gitterdeformationsalgorithmus zur Strömungsberechnung um komplexe Konfigurationen auf Hexaedernetzen. Doctoral thesis, RWTH Aachen University (2006)
- [17] Jaiman, R.K., Jiao, X., Geubelle, P.H., Loth, E.: Assessment of Conservative Load Transfer for Fluid-Solid Interface with Non-Matching Meshes. *Int. J. Numer. Methods Eng.* 55 (2004)
- [18] Kroll, N., Rossow, C.-C., Becker, K., Thiele, F.: The MEGAFLOW Project. *Aerosp. Sci. Technol.* 4(4), 223–237 (2000)
- [19] Kroll, N., Fassbender, J.K. (eds.): MEGAFLOW—Numerical Flow Simulation in Aircraft Design. Springer (2005)
- [20] Kroll, N. (ed.): MEGADESIGN and MegaOpt—German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design. Springer (2009)
- [21] Massjung, R.: Numerical Schemes and Well-Posedness in Nonlinear Aeroelasticity. Doctoral thesis, RWTH Aachen University (2002)
- [22] Park, K.C., Felippa, C.A., Ohayon, R.: Partitioned Formulation of Internal Fluid-Structure Interaction Problems via Localized Lagrange Multipliers. *Comput. Methods Appl. Mech. Eng.* 190, 2989–3007 (2001)
- [23] Quaranta, G., Masarati, P., Mantegazza, P.: A Conservative Mesh-Free Approach for Fluid-Structure Interface Problems. In: *Int. Conf. Comp. Methods Coupled Probl. Sci. Eng.*, Barcelona, Spain (2005)
- [24] Reimer, L., Boucke, A., Ballmann, J., Behr, M.: Computational Analysis of High Reynolds Number Aero-Structural Dynamics (HIRENASD) Experiments. In: *Int. Forum Aeroel. Struct. Dyn. (IFASD) 2009*, paper IFASD-2009-130, Seattle, USA (2009)
- [25] Reimer, L., Braun, C., Wellmer, G., Behr, M., Ballmann, J.: Development of a Modular Method for Computational Aero-Structural Analysis of Aircraft. In: Schröder, W. (ed.) *Summary of Flow Modulation and Fluid-Structure Interaction Findings—Results of the Collaborative Research Center SFB 401 at the RWTH Aachen University, Aachen, Germany*. Springer (1997-2008) (to be published)
- [26] van Rossum, G., Drake, F.L.: Using Python Release 3.1.1. Python Software Foundation (2009), <http://www.python.org>
- [27] Sadeghi, M., Liu, F., Lai, K.L., Tsai, H.M.: Application of Three-Dimensional Interfaces for Data Transfer in Aeroelastic Computations. In: 22nd Appl. Aerodyn. Conf. Exhib., AIAA paper 2004-5376, Providence, USA (2004)
- [28] Schröder, W. (Ed.): *Summary of Flow Modulation and Fluid-Structure Interaction Findings — Results of the Collaborative Research Center SFB 401 at the RWTH Aachen University, Aachen, Germany*. Springer (1997-2008) (to be published)
- [29] Unger, R., Haupt, M.C., Horst, P.: Coupling Techniques for Computational Nonlinear Transient Aeroelasticity. In: 25th Int. Congr. Aeronaut. Sc. (ICAS), paper ICAS 2006-10.2.3, Hamburg, Germany (2006)
- [30] Wendland, H.: Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree. *Adv. Comput. Math.* 4, 389–396 (1995)

Improved Mesh Deformation

Holger Barnewitz and Bernd Stickan

Abstract. An improved, robust, error reducing CFD-mesh deformation module for the parallel simulation environment FlowSimulator is presented. The mesh deformation method is based on radial basis function interpolation for the surface- and volume- mesh nodes combined with a group-weighting and displacement-blending approach. Since the latter weighting and blending approaches are based on given wall distances to the group surfaces, another module for the wall distance computation is introduced. Due to performance reasons, the number of input data locations (base points) used for the radial basis function interpolation must be limited. Therefore, methods have been developed to reduce the number of base points while keeping the interpolation error as low as possible. Furthermore, the modules have been parallelized for usage in multi-node high performance computing clusters. Finally, the capability of a multidisciplinary, parallel application is demonstrated in FlowSimulator with reduced errors and uncertainties.

1 Introduction

Airbus strategy to essentially move much more towards simulation makes it indispensable to know about any uncertainties and deficiencies in the predictive capabilities used for aerodynamic development. Knowing about error bands, their quantity and having in hand some means to manage and minimize their influence on the predicted results could tremendously help in the development process, reliable optimization of the product, shortening of development time and cost.

Holger Barnewitz · Bernd Stickan
AIRBUS Operations GmbH,
Airbus-Allee 1, D-28199 Bremen
e-mail: [holger.barnewitz,bernd.b.stickan}@airbus.com](mailto:{holger.barnewitz,bernd.b.stickan}@airbus.com)

The MUNA project is an essential brick within the Airbus strategy of flight physics/aerodynamics focusing on providing adequate tools for numerical qualification of aerodynamic design during concept phase. Respectively qualified CFD is expected to form the single basis for judgement of aerodynamic status before entering concentrated high level wind tunnel testing – to be ready for next new aircraft development. In addition, MUNA is contributing to support aerodynamic data process change towards "more simulation, less testing".

The contribution described in the following sections focuses on CFD mesh deformation used in the context of numerical aerodynamic shape optimization and shape design including static wing deformation. Major topics are:

- New mesh deformation module "FSDeformation" with advanced methods and integration into FlowSimulator [11]
- Geometry parametrization with a link between CAD (CATIA V5) and mesh deformation
- Use of mesh deformation for aerodynamic shape optimization
- Application of mesh deformation in a CFD/CSM coupled iterative process
- Combining shape design and CFD/CSM coupling in a multi-disciplinary optimization

A typical multi-disciplinary optimization (MDO) process chain for shape optimization of a wing including the static deformation is shown in Fig. 1. An essential brick is the mesh deformation tool which is applied to:

1. reflect the changed shape design generated by a parametric CAD model,
2. deform the wing according to aerodynamic (and other) forces.

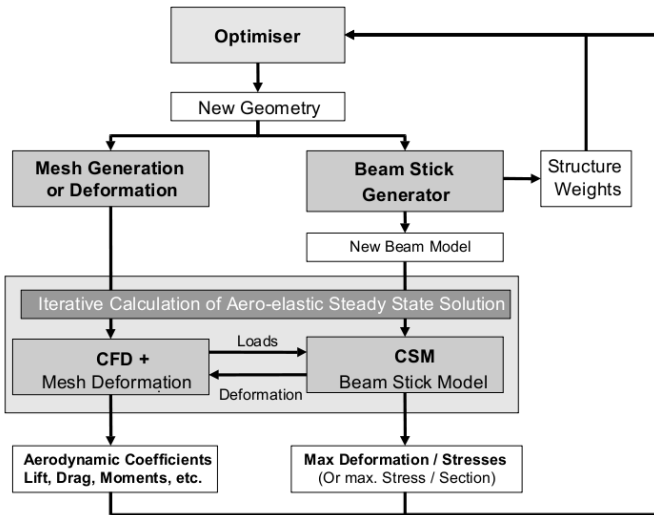


Fig. 1 Multi-disciplinary optimization chain for wing shape and structure weight

The advantage of using mesh deformation for *unstructured grids* is manifold:

- It avoids the problem of *numerical noise* for the calculated aerodynamic coefficients which might occur if new meshes are created for slightly changed geometries. This "noise" is caused by the change of mesh topology. Mesh deformation conserves the topology and small geometry variations produce small mesh deformations in a continuous way.
- A so-called *restart* capability of the flow solver allows to start from a flow solution calculated beforehand to save computing time.
- Usually, deformation of an unstructured CFD mesh is faster than re-generating a new mesh, and thus also saves computing time.

Mesh deformation plays a key role in aerodynamic shape optimization, since any adjustment of the model geometry has an impact on the 3D CFD-mesh. Because CFD-simulations usually rely on spatial discretization based on volume-meshes, these have to be updated if a CAD surface changes its location or its shape. The costs for the generation of new meshes should be as low as possible but should also produce usable grids even for large local changes in the model geometry. There are several possible ways to update the mesh, e.g. the complete re-meshing of the complete grid or the deformation of an existing grid.

For unstructured meshes, re-meshing would certainly change the topology of the mesh. Since the discretization in 3D space is generally not dense enough to produce a mesh-independent flow solution, the newly generated mesh would produce a slightly different flow field not caused by the geometry change. This leads, especially in the case of shape optimization, to *noise in the aerodynamic coefficients*, which often significantly disturbs the shape optimizer. Furthermore, the computational cost for re-meshing of unstructured grids is very high. Hence, mesh deformation is an essential tool in this area of computational fluid dynamics:

- the topology of the mesh remains unchanged and
- small geometry variations correspond to small changes of the *numerically* determined aerodynamic flow field.

2 CFD Mesh Deformation Module

The deformation module "FSDeformation" has been developed for the simulation environment FlowSimulator [1]. It is based on an implementation of the mesh-deformation module presented in [6] using the radial basis function (RBF) interpolation approach. This approach is extended by the feature of specifying groups of different boundaries with separate interpolation functions and a blending function, which restricts the deformation to a specified zone around these boundaries. Both features are controlled by the distances d of the mesh nodes to the group target boundaries. The distances are calculated by the wall distance module (section 3).

More details about the deformation module, for example concerning parallel performance and interpolation quality, can be found in [10].

2.1 Radial Basis Functions in Mesh Deformation

Deformation methods based on RBF-interpolation are independent of the volume mesh and flow solver type, because the algorithm is working on completely arbitrary clouds of points without using any connectivity information. Additionally, for the mesh updates of consecutive optimization steps or an unsteady aeroelastic simulation, only a matrix-vector multiplication is necessary for each mesh node. The computationally most expensive part is to compute the interpolation matrix for this multiplication. It can be calculated once in the beginning of the entire simulation and remains unchanged, since it only depends on the base points, but not on the deformation vectors. Consequently, the method can be perfectly parallelized (using MPI and partitioned grids), because each process has to apply the interpolation matrix only to its own grid nodes. But it is also clear that the dimension of the interpolation matrix highly influences the overall speed of the algorithm.

2.1.1 Multivariate Interpolation Using Radial Basis Functions

The radial basis functions approach is a well-established interpolation method for gridded and scattered data, whereas the most natural context for function approximation is given for scattered data [3, p. 99], [4, p. 4]. In the field of computational fluid dynamics (CFD) it is often used for coupling CFD-grids to finite element structure grids.

The input data in d dimensions consist of data locations \mathbf{x}_i , merged into the data-set

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathbb{R}^d, \quad (1)$$

and the corresponding function values

$$f_i = f(\mathbf{x}_i) \in \mathbb{R}, \quad i = 1, \dots, n. \quad (2)$$

The data locations \mathbf{x}_i are called centers or "base points".

The goal is to interpolate the function values between the base points by an approximant $s : \mathbb{R}^d \rightarrow \mathbb{R}$ to satisfy the condition

$$s|_X = f|_X. \quad (3)$$

In this specific case s is a linear combination of shifted radially symmetric basis functions ϕ . Radially symmetric means that the value of $\phi(\cdot)$ depends only on the distance of the argument to the origin, hence it is often written $\phi(\|\cdot\|)$. The distance norm is usually the Euclidean norm (with $d = 3$)

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^d x_i^2}. \quad (4)$$

$s(\mathbf{x})$ has the general form

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|). \quad (5)$$

Setting $s(\mathbf{x}_i)$ equal to f_i for all $i \in \{1, \dots, n\}$ leads to the linear system

$$A\mathbf{y} = \mathbf{b} \quad (6)$$

with

$$A = (\phi(\|\mathbf{x}_j - \mathbf{x}_k\|))_{(j,k)=1,\dots,n}, \quad \mathbf{y} = (\alpha_i)_{i=1,\dots,n}, \quad \mathbf{b} = (f_i)_{i=1,\dots,n}. \quad (7)$$

A unique interpolant is usually (for most ϕ) guaranteed, if the base points are all distinct and there are at least two of them [3, p. 6]. An example for a radial basis function could be $\phi(\|\mathbf{x}\|) = \|\mathbf{x}\|^2 \log \|\mathbf{x}\|$, which is called "*thin plate spline*".

An important attribute of this interpolation method is the possibility to expand the approach of equation (5) by adding a polynomial to the definition without losing the uniqueness of the coefficients. For function values f_i , which show a polynomial character, the appended polynomial improves the interpolation quality. The only restriction is that the polynomial must have a degree $m \geq 1$ and is non-zero at all base points. This leads to:

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}). \quad (8)$$

The coefficients can be computed by solving

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}) = f_i \quad (9a)$$

$$0 = \sum_{i=1}^n \alpha_i q(\mathbf{x}_i) \quad \forall q: \deg(q) \leq \deg(p) \quad (9b)$$

The extra equation (9b) takes up the extra degrees of freedom given by the polynomial coefficients, to allow a unique interpolant. The uniqueness can be guaranteed, if ϕ is "conditionally positive definite". It is referred to [3, p. 101] for more details to the theory of this topic.

Again, the requirements on X are not very strong. For a linear polynomial, X must only contain four base points, which do not lie on a plane. Furthermore, if the function values f_i at the base points were generated by a linear function, they would be reproduced exactly by the linear polynomial. [4, p. 5]

In the following the dimension is set to $d = 3$ in this document. Since $\mathbf{x} = (x_x, x_y, x_z)$, the polynomial is linear and can be written as

$$p(\mathbf{x}) = \beta_0 + \beta_1 x_x + \beta_2 x_y + \beta_3 x_z = \boldsymbol{\beta}^T \begin{pmatrix} 1 \\ \mathbf{x} \end{pmatrix}. \quad (10)$$

So equations (9) can be abstracted to matrix notations

$$H\mathbf{y} = \mathbf{b}, \quad (11)$$

with

$$A = (\phi(\|\mathbf{x}_j - \mathbf{x}_i\|))_{(j,i)=1,\dots,n} \in \mathbb{R}^{n \times n}, \quad (12)$$

$$P = \left(\begin{array}{c} 1 \\ \mathbf{x}_k \end{array} \right)_{k=1,\dots,n} \in \mathbb{R}^{4 \times n}, \quad (13)$$

$$H = \begin{pmatrix} 0 & P \\ P^T & A \end{pmatrix} \in \mathbb{R}^{(n+4) \times (n+4)}, \quad (14)$$

$$\mathbf{y} = \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} (\beta_i)_{i=1,\dots,4} \\ (\alpha_i)_{i=1,\dots,n} \end{pmatrix} \in \mathbb{R}^{n+4} \text{ and} \quad (15)$$

$$\mathbf{b} = \begin{pmatrix} 0 \\ \mathbf{f} \end{pmatrix} = \begin{pmatrix} 0 \\ (f_i)_{i=1,\dots,n} \end{pmatrix} \in \mathbb{R}^{n+4}. \quad (16)$$

Solving (11) provides in \mathbf{b} the coefficients to use (8) for the interpolation of arbitrary points. The matrix H will be called "interpolation matrix" below, although it is only used to calculate the interpolation coefficients.

The module presented is not independent of cell connectivity, since wall distances of the nodes to certain boundary groups are used. The algorithm to compute the wall distances relies on connectivity information to determine neighboring nodes. But, it is important to note that the base points $\mathbf{x}_{s,i}$, $i = 1, \dots, n_s$ do not need any connectivity information.

Solutions for the indicated performance factor "interpolation matrix size", which directly depends on the number of used input deformation vectors, will be shown in section 4. That section contains different methods to reduce the number of base points and deformation vectors.

The basic interpolation functions of the module are taken from DLR's flow solver TAU. They have been applied successfully at DLR and Airbus to many test cases.

2.2 Algorithm

The interpolation algorithm is based on a *group-weighting* and a *deformation-blending* approach.

A group-weighting approach is used to allow the independent movement of different model parts/boundaries in the grid. Otherwise the deformations of different boundaries could influence each other and unintentional surface deformation would be the result. Separating the interpolation by group protects the shape of the different bodies. Therefore, the interpolation matrix H^g of each group g has to be computed

and applied to the grid nodes separately. Finally, the deformation result for each grid point is calculated by a weighted average of each group-deformation result.

The deformation-blending approach supports the protection of boundary layer cells and the usage of radial basis functions $\phi(\|x\|)$ with limits $\phi(\|x\|) \rightarrow \infty$ for $\|x\| \rightarrow \infty$. These radial basis functions, which increase with increasing distance to the base point of a deforming body, need to be restricted farther away from the surface of this body. Otherwise local deformations would influence the whole mesh. Additionally, the added polynomial of the interpolation approach (8) would deform the whole volume mesh as well. Consequently, this approach that is implemented to recover linear deformations exactly, cannot be used without the blending of deformation values.

Hence, the notations are expanded by an elevated group index g for n_g groups. As input data there are n_s^g base points $\mathbf{x}_{s,i}^g \in \mathbb{R}^3$ for each group g merged into the datasets

$$X_s^g = \left\{ \mathbf{x}_{s,1}^g, \mathbf{x}_{s,2}^g, \dots, \mathbf{x}_{s,n_s^g}^g \right\} \quad \text{for } g = 1, \dots, n_g. \quad (17)$$

The function values that are going to be interpolated, are the deformation vectors

$$\Delta \mathbf{x}_{s,i}^g = \Delta \mathbf{x}(\mathbf{x}_{s,i}^g) = \begin{pmatrix} \Delta x_{s,i}^{g,x} \\ \Delta x_{s,i}^{g,y} \\ \Delta x_{s,i}^{g,z} \end{pmatrix} \in \mathbb{R}^3 \quad \text{for } i = 1, \dots, n_s^g, \quad g = 1, \dots, n_g, \quad (18)$$

which could be used to compute the displaced coordinates $\mathbf{x}_{\text{new},i}^g$ of the base points:

$$\mathbf{x}_{\text{new},i}^g = \mathbf{x}_{s,i}^g + \Delta \mathbf{x}_{s,i}^g \quad \text{for } i = 1, \dots, n_s^g, \quad g = 1, \dots, n_g. \quad (19)$$

But the aim of the deformation module is to update the mesh nodes and not the base points.

A difference to the function values f_i in equation (2) to the function values $\Delta \mathbf{x}_{s,i}^g$ is their dimension. Section 2.1.1 only deals with one-dimensional function values while in this case the function values are three-dimensional. Therefore each coordinate of the mesh nodes has to be interpolated separately. It is advantageous that the interpolation matrix H^g in (11) has to be computed only once for each boundary group instead of computing it for each dimension separately, since the matrix depends only on the base points $\mathbf{x}_{s,i}$ and the chosen radial basis function ϕ . So the interpolation matrices H^g for each group can be stated as:

$$H^g = H(X_s^g, \phi). \quad (20)$$

For each dimension $k \in \{x, y, z\}$ the interpolation coefficients $\boldsymbol{\alpha}^{g,k} = \left(\alpha_i^{g,k} \right)_{i=1, \dots, n_s^g}$ and $\boldsymbol{\beta}^{g,k} = \left(\beta_i^{g,k} \right)_{i=1, \dots, 4}$ can be calculated by inverting equation (11):

$$\begin{pmatrix} \boldsymbol{\beta}^{g,k} \\ \boldsymbol{\alpha}^{g,k} \end{pmatrix} = (H^g)^{-1} \begin{pmatrix} \mathbf{0} \\ \left(\Delta \mathbf{x}_{s,i}^{g,k} \right)_{i=1, \dots, n_s^g} \end{pmatrix}. \quad (21)$$

The actual interpolation algorithm calculates the deformations of the grid nodes

$$d\mathbf{x}_{v,i} = (dx_{v,i}^x \ dx_{v,i}^y \ dx_{v,i}^z)^T \tag{22}$$

for the volume mesh grid nodes $\mathbf{x}_{v,i}$ by using the distance d_i^g to the nearest surface of group g . For every coordinate $k \in \{x, y, z\}$ the governing equations are:

$$dx_{v,i}^{g,k} = \sum_{j=1}^{n_s^g} \alpha_j^{g,k} \phi(\|\mathbf{x}_{v,i} - \mathbf{x}_{s,j}\|) + (\boldsymbol{\beta}^{g,k})^T \begin{pmatrix} 1 \\ \mathbf{x}_{v,i} \end{pmatrix}, \quad g = 1, \dots, n_g \tag{23}$$

$$\text{blend}(d_i^g, g) = \begin{cases} 0 & : d_i^g > \text{RZW}^g \\ 1 & : d_i^g < \text{RFW}^g \\ \frac{\text{RZW}^g - d_i^g}{\text{RZW}^g - \text{RFW}^g} & : \text{else} \end{cases} \tag{24}$$

$$\text{weight}(d_i^g) = \frac{1}{\sqrt{\max\{d_i^g, \varepsilon\}}} \tag{25}$$

$$dx_{v,i}^k = \frac{\sum_{g=1}^{n_g} \text{blend}(d_i^g, g) \cdot \text{weight}(d_i^g) \cdot dx_{v,i}^{g,k}}{\sum_{g=1}^{n_g} \text{weight}(d_i^g)} \tag{26}$$

Two new functions have been introduced: the blending function $\text{blend}(\cdot)$ and the weighting function $\text{weight}(\cdot)$. The weighting function averages the individual group deformations. Because its limit for $d_i \rightarrow 0$ is infinity, it needs a cut-off value $1/\sqrt{\varepsilon}$ for numerical reasons.

The blending function is sketched in figure 2. With its group-parameters RZW^g (Radius Zero Weight) and RFW^g (Radius Full Weight) it is controlling the deformation of the grid nodes. If a grid node is close to a boundary of group g with a distance less than RFW^g , it will move approximately like the boundary. This functionality can be used to conserve the sensitive boundary layer part of a grid. Farther away from the boundary with a distance $d_i^g > \text{RZW}^g$ the deformation is zero.



Fig. 2 Blending function for grid node deformation computation, including the parameter radius full weight (RFW) and radius zero weight (RZW)

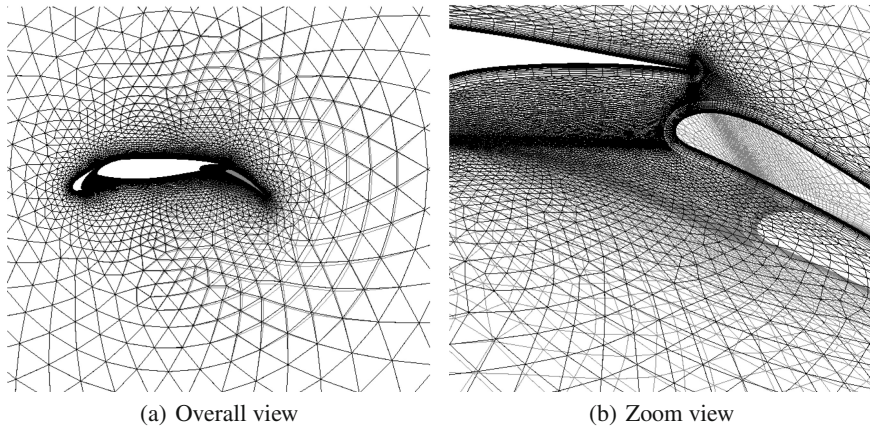


Fig. 3 2d test case, wing including flap and slat. Each of the 3 parts is an independent deformation group and only the flap has input values unequal to zero (undeformed: black, deformed: grey)

An example for independently deforming groups can be seen in figure 3. It shows that the surface mesh of the rigid main wing body is not affected by the deformation of the nearby moving flap. The radius zero weight can be recognized in figure 3(a), too.

The algorithm is also described shortly in [6]. This paper also provides test cases showing the usefulness of the presented group-weighting approach and the quality conserving capability of the methodology.

Several different boundary type dependent algorithms have been developed to simplify the usage of standard cases often applied to CFD meshes for aircrafts:

- *Standard Boundary Type*. This is handled as described above. Deformation vectors have to be provided for this surface type.
- *No-Normal-Movement Boundary Type*. All surface points on this boundary are allowed to slide on the surface. The movement in surface normal direction is suppressed. It's used for example for symmetry planes.
- *Far-field Boundary Type*. Here the deformation is set to zero.
- *Attached Group Boundary Type*. This treatment conserves the shape of an attached device, e.g. an engine mounted on a deforming wing.

3 Wall Distance Module

The mesh deformation module presented in section 2 uses the distance of the grid points to the closest group boundary for the weighting of groups and for the blending of deformations. The wall distances control the influence of different boundary groups on the deformation of a specific volume grid node.

There are different approaches for the computation of wall distances, for example based on partial differential equations, as seen in [11], based on a clever merging of the boundary points [13], or just a naive brute force algorithm, which compares each boundary node with every volume mesh node. The presented method, which was adopted from DLR's TAU preprocessor, uses another algorithm. In TAU the wall distance is used for the application of certain turbulence models. The method is, like the mesh deformation module, embedded into an independent module for the simulation environment FlowSimulator.

The algorithm uses an advancing frontier approach. Every grid node n gets an additional parameter $\mathbf{x}_{\text{near}}[n]$, which saves the coordinates of the currently nearest boundary node. Then in every iteration step, each node compares the distance to its \mathbf{x}_{near} -entry with the \mathbf{x}_{near} values of its neighboring nodes and where required updates the \mathbf{x}_{near} -value with a better value from a neighbor. Since the boundary nodes have the correct solution directly at the beginning, the solution for \mathbf{x}_{near} for each node moves into the field node by node.

This so-called advancing front algorithm makes it possible that for certain (structured) grids it would take only a few iteration steps to advance the correct solution for \mathbf{x}_{near} into the interior of the grid.

4 Base Point Reduction Methods

The number of base points n_s has a major influence on the performance of the radial basis function interpolation algorithm. The needed (direct) matrix inversion depends on the third power of n_s and the interpolation of the grid points depends linearly on the base point number. If the tool is used for the coupling of a structural finite elements (FEM) grid to a computational fluid dynamics grid, the number of input base points will be equal to the number of surface grid nodes of the FEM-grid. The common number of surface nodes of these grids is way too large to use them all for the RBF grid deformation and still having satisfactory runtime results. So the reduction of the base points is indispensable for the mesh deformation module.

The reduction of the base points is not the only way to increase the efficiency of radial basis function interpolation methods. Other possibilities are, for example, multilevel approaches combined with base point reduction [9] or partition of unity approaches like in [12]. The multilevel approach uses a base point set hierarchy to start the interpolation at a coarse level and then refining it progressively. The partition of unity approach breaks the large problem down to several small ones by partitioning the base points into neighbor sets.

A useful attribute of the radial basis function interpolation approach is that no connectivity information of the input base points is needed. To conserve this characteristic the reduction algorithms do not use connectivity information as well.

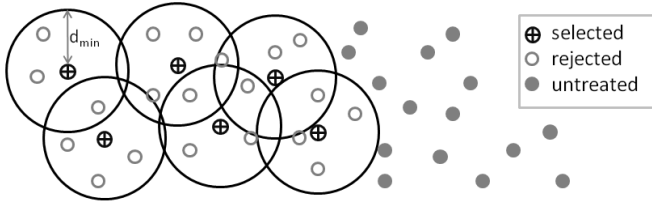


Fig. 4 Equidistant reduction snapshot during iteration step

4.1 Equidistant Reduction Method

The TAU deformation module also contains this method to reduce the number of base points. It tries to select the base points X_s spatial-evenly distributed from the point set X_{inp} . This is managed by iteratively finding the right minimal distance d_{min} to possible neighboring base points, to get as close as possible to the maximal number of desired base points $n_{s,max}$. Neighbors with a distances less than d_{min} are rejected during this process. Due to performance issues, it is using an octree data structure to find the neighbors closer than d_{min} to a specified base point. Figure 4 sketches one iteration step of the algorithm.

The result is having evenly distributed base points. But choosing the base points like this does not take into account the deformation vectors $\Delta x_{inp,i}$ or *interpolation errors*.

4.2 Weighted Distances

This approach is similar to the equidistant reduction approach of section 4.1. The idea is to modify the distances between two input sites $x_{inp,i}$ and $x_{inp,j}$ by a weighting factor $w_{i,j}$. The consequence would be an increased point density in areas of higher weights.

The distance $d_{i,j}$ between two points x_i and x_j with associated weights w_i and w_j is calculated by

$$d_{i,j} = \frac{w_i + w_j}{2} \|x_i - x_j\|_2. \quad (27)$$

The disadvantage of this approach is that the octree data structure of section 4.1 cannot be used any more. Instead a simple list data structure has to be used. Searching for neighbors of one node will therefore include to check the distance to all input sites X_{inp} .

4.2.1 Weighting by the Difference to the Local Average Deformation

This weighting approach takes the deformation vectors $\Delta x_{\text{inp},i}$ into account directly. It uses a function $FindNeighbours(\mathbf{x}, X, \Delta X, d)$, which returns subsets of

$$X_{\text{nb}} = \{x_{\text{nb},1}, x_{\text{nb},2}, \dots, x_{\text{nb},n_{\text{nb}}}\} \quad (28)$$

and

$$\Delta X_{\text{nb}} = \{\Delta x_{\text{nb},1}, \Delta x_{\text{nb},2}, \dots, \Delta x_{\text{nb},n_{\text{nb}}}\} \quad (29)$$

of cardinality n_{nb} with points of distance less than d to \mathbf{x} . In this case the distances are not weighted yet. The weights for the input point $x_{\text{inp},i}$ are then calculated by

$$(X_{\text{nb}}, \Delta X_{\text{nb}}) = FindNeighbours(x_{\text{inp},i}, X_{\text{inp}}, \Delta X_{\text{inp}}, d) \quad (30)$$

$$\overline{\Delta x_{\text{nb}}} = \frac{\sum_{i=1}^{n_{\text{nb}}} \|\Delta x_{\text{nb},i}\|_2}{n_{\text{nb}}} \quad (31)$$

$$w_i = \frac{|\overline{\Delta x_{\text{nb}}} - \|\Delta x_{\text{inp},i}\|_2|}{\sum_{i=1}^{n_{\text{inp}}} |\overline{\Delta x_{\text{nb}}} - \|\Delta x_{\text{inp},i}\|_2|} \quad (32)$$

The idea is to develop an expression that favors the base points, the absolute deformation value of which is different to the average deformation value $\overline{\Delta x_{\text{nb}}}$ of its neighbors. Another thought is that the nodes at the outer tips of a deforming body will get a higher weight, since at the tip the deformations reach usually their maximum and consequently differ strongly from the neighborhood mean. An approach which only takes the gradient into account would not result in a higher weight for the outer base points, because the deformation gradient would not have a peak at an outer base point. An example to illustrate this idea can be seen in figure 5. This simplified example shows why the approach leads to higher weights for the base points on the tip, the deformation vectors of which should not be neglected in the final base point set. It shows a tip body with a slight rotational deformation. The deformation vector with the largest value is on the tip of the body. The equidistant reduction algorithm from section 4.1 could easily fail to select the maximum deformation vectors. Because of the higher weight values at the tip this would happen less likely with the new algorithm.

This example shows a disadvantage of the algorithm as well. If the gradient of the deformation vectors is constant in a certain area, the weights will tend to zero. This will lead to a very low base point density in the next step. To get a lower border for the density, the final reduction method is combined with the equidistant reduction method. First a fraction $\text{frac}_{\text{equi}}$ of the desired $n_{s,\text{max}}$ base points is chosen by the equidistant algorithm, then the remaining base points and deformation vectors are selected with the weighted distance approach.

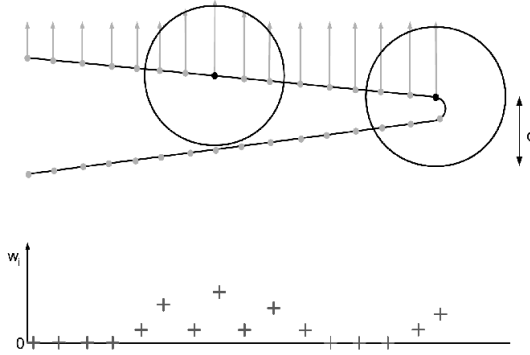


Fig. 5 Schematic example for weights w_i of the upper base points with their deformation vectors

4.2.2 Weighting by Interpolation Error

The algorithm presented in this section is combining the approach of the weighted distance reduction with the interpolation error calculated for the input data locations. In this case, the distance weights

$$W = \{w_1, w_2, \dots, w_{\text{inp}}\} \tag{33}$$

are equal to the error of interpolated deformation vectors $\Delta\tilde{X}_{\text{inp}}$.

The basic scheme of the algorithm looks like:

- Select start base point set X_s with corresponding deformation vectors ΔX_s by equidistant reduction
- Do n_{EWSteps} times:
 - Interpolate deformations at input points X_{inp} , by using the sets X_s and ΔX_s , to get the deformation vectors $\Delta\tilde{X}_{\text{inp}}$
 - Calculate weights w_i by comparing ΔX_{inp} to $\Delta\tilde{X}_{\text{inp}}$
 - Add further base points and deformation vectors by weighted distance reduction to X_s and ΔX_s , respectively.

The type of greedy algorithm, which recalculates the exact interpolation error in each step, is also proposed in [2, p.9].

To interpolate the input points X_{inp} in each step, a new interpolation matrix H_k has to be created from the already chosen base points X_s and inverted in every iteration step. Then the error can be calculated by interpolating the deformation vectors ΔX_s of these base points to the input set X_{inp} to get the interpolated data set

$$\Delta\tilde{X}_{\text{inp}} = \left\{ \Delta\tilde{\mathbf{x}}_{\text{inp},1}, \Delta\tilde{\mathbf{x}}_{\text{inp},2}, \dots, \Delta\tilde{\mathbf{x}}_{\text{inp},n_{\text{inp}}} \right\} \tag{34}$$

and taking the pairwise difference to the input deformation vector set ΔX_{inp} to compute the weights

$$w_i = \|\Delta \tilde{\mathbf{x}}_{\text{inp},i} - \Delta \mathbf{x}_{\text{inp},i}\|_2, \quad i = 1, 2, \dots, n_{\text{inp}}. \quad (35)$$

4.3 Error Correction

The error correction algorithm was originally presented in [2, p. 7]. The algorithm tries to correct the $\boldsymbol{\alpha}$ -interpolation coefficient vector locally. Therefore, unlike the previous methods, it is not using the interpolation approach including a polynomial (9), but instead the basic approach without an added polynomial (5):

$$s(\mathbf{x}) = \sum_{i=1}^n \alpha_i \phi(\|\mathbf{x} - \mathbf{x}_i\|). \quad (36)$$

Furthermore, the coefficients α_i are not calculated by inverting the interpolation matrix A , but instead by correcting them during the iterations continuously. In each step the interpolation error $e_i, i = 1, \dots, n_{\text{inp}}$ of all deformation vectors

$$\Delta \tilde{X} = \left\{ \Delta \tilde{\mathbf{x}}_1, \Delta \tilde{\mathbf{x}}_2, \dots, \Delta \tilde{\mathbf{x}}_{n_{\text{inp}}} \right\} \quad (37)$$

at the data sites

$$X_{\text{inp}} = \left\{ \mathbf{x}_{\text{inp},1}, \mathbf{x}_{\text{inp},2}, \dots, \mathbf{x}_{\text{inp},n_{\text{inp}}} \right\} \quad (38)$$

is recalculated. The coefficients

$$\boldsymbol{\alpha} = (\alpha_i)_{i=1, \dots, n_{\text{inp}}} \quad (39)$$

and the deformation vectors $\Delta \tilde{\mathbf{x}}_i$ are adjusted locally by the radial basis function belonging to the base point with the largest interpolation error $e_{i_{\text{worst}}} = \|\text{d}\Delta \mathbf{x}_{i_{\text{worst}}}\|_2$. The correction of $\alpha_{i_{\text{worst}}}$ is performed by

$$\Delta \alpha_{i_{\text{worst}}} = \frac{1}{\phi(0)} \text{d}\Delta \tilde{\mathbf{x}}_{i_{\text{worst}}}, \quad (40)$$

which is used to update the interpolation values of all base points by

$$\Delta \tilde{\mathbf{x}}_i = \Delta \tilde{\mathbf{x}}_i + \Delta \alpha_{i_{\text{worst}}} \phi(\|\mathbf{x}_i - \mathbf{x}_{i_{\text{worst}}}\|_2) \quad (41)$$

In every step in equation (41) the error $e_{i_{\text{worst}}}$ at of the deformation $\Delta \tilde{\mathbf{x}}_{i_{\text{worst}}}$ is changed to zero, since

$$\begin{aligned} \Delta \tilde{\mathbf{x}}_{i_{\text{worst}}} &= \Delta \tilde{\mathbf{x}}_{i_{\text{worst}}} + \frac{\phi(\|\mathbf{x}_{i_{\text{worst}}} - \mathbf{x}_{i_{\text{worst}}}\|_2)}{\phi(0)} d\Delta \tilde{\mathbf{x}}_{i_{\text{worst}}} \\ &= \Delta \tilde{\mathbf{x}}_{i_{\text{worst}}} + \Delta \mathbf{x}_{i_{\text{worst}}} - \Delta \tilde{\mathbf{x}}_{i_{\text{worst}}} = \Delta \mathbf{x}_{i_{\text{worst}}}. \end{aligned} \tag{42}$$

But the corrections for the deformation vectors $\Delta \tilde{\mathbf{x}}_i$, which are located inside the impact area of the base point $\mathbf{x}_{i_{\text{worst}}}$, are not necessarily decreasing the interpolation error. Hence, if $n_{s,\text{max}} > n_{\text{inp}}$ base points should be selected, the algorithm will run infinitely without reducing the interpolation error to zero. Additionally, the algorithm "tends to show a degree of self limiting behavior in terms of how many points it uses ([...]), often returning to correct a previously identified point rather than introducing a new one" [2, p. 8].

The paper [2] uses the algorithm above to approximate the coefficients α_i of equation (36), but also recommends not to use these coefficients. Instead the selected base points in X_s should be used for exact interpolation, which uses the inversion of the interpolation matrix as seen in section 2.1.1. The algorithm implemented into the deformation module presented in this document uses this recommendation and, secondly, instead of the interpolation approach (36) the approach including a polynomial as seen in equation (8) for the interpolation matrix creation.

Because the results were still not satisfactory, this approach has been combined with an initial equidistant reduction step to choose $\text{frac}_{\text{equi}} \cdot n_{s,\text{max}}$ base points by the algorithm presented in section 4.1.

A big disadvantage of the algorithm is that it only works with radial basis functions $\phi(r)$ with the maximal value for $r = 0$, so radial basis functions with local influence range.

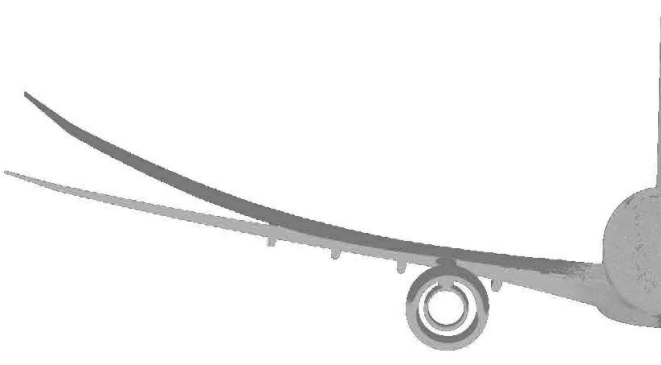


Fig. 6 Half model test case

5 Interpolation Quality Comparison

The different reduction algorithms in section 4.1 to 4.3 select different base point sets X_s from the input data site set X_{inp} . This section is comparing the resulting interpolation errors in a test case.

Therefore the extremely deformed half model airplane, as seen in figure 6 is used. The input base points X_{inp} and their deformation vectors ΔX_{inp} were calculated with a structural loads program. The tool generates for each surface grid node a deformation vector, so the cardinality of X_{inp} and ΔX_{inp} is quite large with a value of $n_{inp} = 137, 136$ for the wing without engine only.

Figure 7 shows the interpolation error e_i for the bottom surface of the wing, because it used to show higher interpolation errors. The settings for the interpolation and base point reduction can be seen in table 1.

Because the structural loads tool gives an deformation output for every surface grid node, the interpolation error for the surface nodes $\mathbf{e} = (e_i)_{i=1, \dots, n_{inp}}$ can be calculated by taking the differences between the calculated interpolations $\Delta \tilde{\mathbf{x}}_i$ and the input deformations $\Delta \mathbf{x}_i$:

$$e_i = \|\Delta \tilde{\mathbf{x}}_i - \Delta \mathbf{x}_i\|_2. \tag{43}$$

The picture 7(a) clearly shows that the base points, chosen by the equidistant reduction method, are not satisfactory in the outer wing part. The outer 30 percent of the wing show strongly increased error values. This result has motivated to improve the base point selection process. The other reduction algorithms show a strongly improved interpolation error in this part of the wing, too. The mean absolute error \bar{e} of each test case for the actual 2000 base point setting, and additionally for another test series with only 1000 base points is given in table 2. Furthermore, this table contains the variance $Var(\mathbf{e})$ and the maximal error $\max_i(e_i)$.

The table confirms the impression of the given plots: All new methods choose base points resulting in a significantly lower interpolation error. Furthermore, the variance $Var(\cdot)$ indicates that less fluctuations in the error can be expected. The maximum error is lowered by up to 92 percent.

Figure 8 shows how the base points are chosen by the different algorithms. The equidistant reduction algorithm (7(a)) distributes the base points nicely over the whole domain. Taking a closer look, the decreased density of points in the thin parts of the wing, like the trailing edge and the tip, can be recognized.

Table 1 Test settings

Reduction method	Parameter	RBF ϕ
Equidistant reduction	-	Wendland's C^0 , impact radius $r = 20.0$
Local average weighting	$\text{frac}_{\text{equi}} = 0.5, d_{\text{min}} = d_{\text{max}}/10$	
Error weighting	$\text{frac}_{\text{equi}} = 0.5, n_{\text{EWSteps}} = 3$	
Error correction	$\text{frac}_{\text{equi}} = 0.5$	

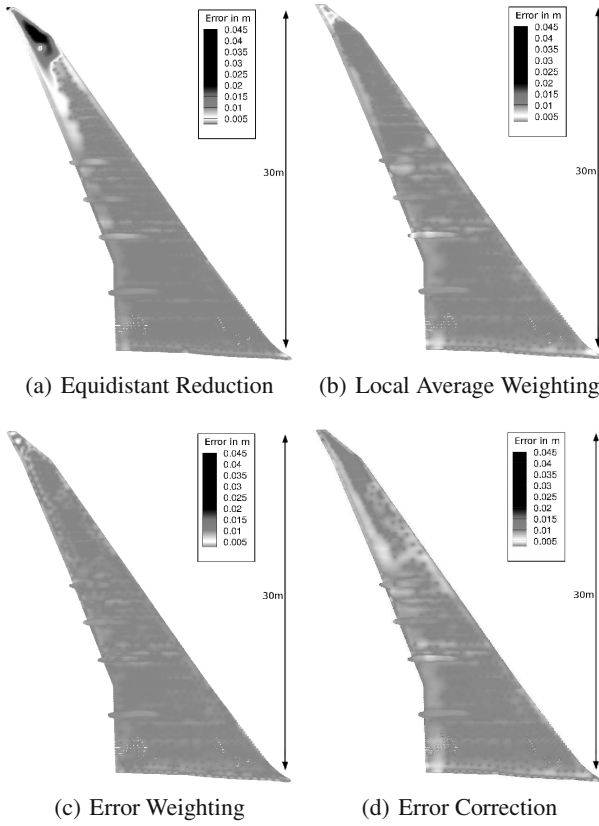


Fig. 7 Interpolation error of wing, lower surface view, 2000 base points, color table: absolute interpolation error in m

Table 2 Test results interpolation error $\mathbf{e} = (e_i)_{i=1, \dots, n_{\text{inp}}}$

(a) 1000 base points

<i>Reduction method</i>	$\bar{e} [m]$	$Var(\mathbf{e}) [m^2]$	$\max_i(e_i) [m]$
Equidistant reduction	5.30E-03	8.52E-02	1.24E-04
Local average weighting	1.56E-03	3.87E-02	3.83E-06
Error weighting	1.52E-03	1.10E-02	3.00E-06
Error correction	1.40E-03	8.99E-03	1.19E-06

(b) 2000 base points

<i>Reduction method</i>	$\bar{e} [m]$	$Var(\mathbf{e}) [m^2]$	$\max_i(e_i) [m]$
Equidistant reduction	2.33E-03	2.52E-05	4.99E-02
Local average weighting	8.60E-04	1.92E-06	1.91E-02
Error weighting	6.86E-04	6.20E-07	7.92E-03
Error correction	7.38E-04	4.55E-07	3.83E-03



Fig. 8 2000 base points selected by different reduction algorithms

The new methods have all used the equidistant reduction algorithm in the first step for half of their base points. The remaining half has been selected differently, besides all algorithms concentrate the selected points in the outer wing part.

The local average method (8(b)) is the most extreme example for this behavior. Because the deformations are increasing with a parabolic character, the weights used to be more significant in the outer part. Supplementary, the more narrow getting wing supports this behavior, because the neighborhood of a certain point would contain more points of the side closer to the fuselage than from the outer side, which influences the local mean deformation.

The two remaining approaches based on the interpolation error are choosing their base points similarly. A difference between the error based greedy algorithms is that the error weighting algorithm is distributing the points more numerous in areas far away from the outer wing. The error correction algorithm has selected most points in the tip area.

The results show the best point selection for the error weighting and error correction method. But one has to keep in mind that error correction is only working with radial basis functions with limited influence range, while error weighting has the larger computational costs (7 times larger than error correction).

6 Applications

6.1 Wing Shape Design

To accurately compare the aerodynamic coefficients (e.g. drag) between small changes of the aerodynamic shape of a wing, it is necessary to be as independent from the CFD mesh as possible. Otherwise, difficulties arise to distinguish between grid discretization effects ("numerical noise" due to change of grid topology) and geometric effects. Nowadays in industrial context a 3D *unstructured* CFD mesh is not made in a way to obtain a *mesh independent* CFD solution.

However, mesh deformation conserves the grid topology and small geometry variations produce small mesh deformations in a continuous way. Utilizing this, comparisons of aerodynamic coefficients are better possible and thus uncertainties otherwise introduced by changes of grid topology are minimized. Mesh deformation with FSDeformation was here successfully applied to a shape design change for a wing-tip (figure 9).

The discrete deformation field was obtained from the parametric CAD model (CATIA V5) using a two-stage process (first the treatment of curves and then surfaces). In a predictor step, the deformation field is determined by subtracting points on discretized corresponding CAD curves. This gives an initial surface deformation which may not be accurate on the inner region of surface panels apart from the bounding curves.

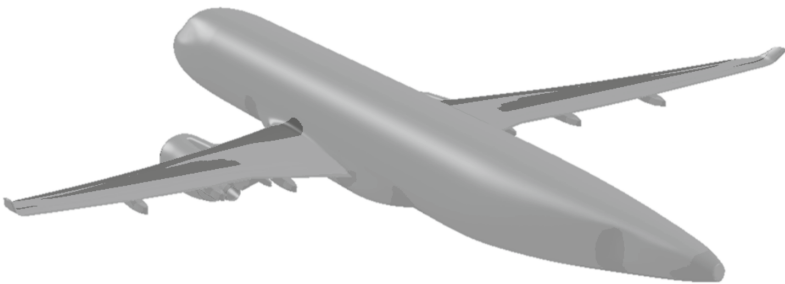


Fig. 9 Wing tip shape design example

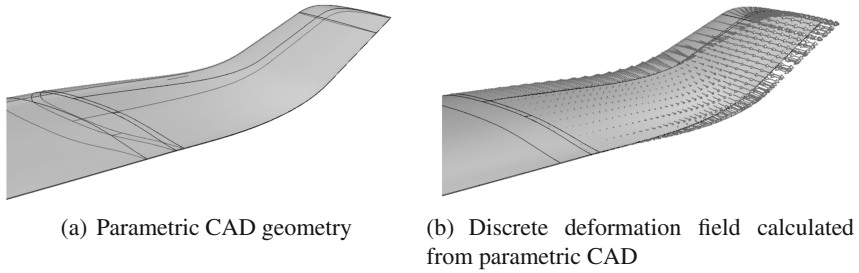


Fig. 10 Wing tip design with deformation obtained from a parametric CAD geometry

In the corrector step surface points are projected to the new CAD geometry. The projection vectors together with the displacement vectors of the discretized curves then builds the final discrete deformation field (figure 10(b)). It serves as input for the new tool FSDeformation to accurately move all points of the CFD grid corresponding to the parametric change of the CAD geometry.

6.2 Multi-disciplinary Wing Optimization—SFB-401-Wing

The task given was to optimize a wing with respect to aerodynamics, structures, and performance under considerations of static aeroelastic effects. The study involves the sizing of the wing box skins and spars to obtain minimum weight fulfilling static aero-elastic requirements (details in [7, p. 287]).

The considered MDO process chain for shape optimization of a wing including the static deformation is shown schematically in Fig. 11. The objective function is:

$$\text{Obj} = W_{A/C} \times C_D / C_L, \quad (44)$$

where $W_{A/C}$ is the total weight of the aircraft, C_D is the overall aerodynamic drag coefficient, and C_L is the aerodynamic lift coefficient. The objective, thrust, is equivalent to the total aerodynamic drag force in stationary horizontal flight, which should be minimized.

A CATIA V5 parametric model of the wing is controlled by the optimizer using an external CATIA-DesignTable, where all relevant shape parameters for the wing are listed. The shape of airfoils at four predefined wing sections (root, kink1, kink2, and tip section) can be changed parametrically to control the thickness, camber, and twist distribution of the wing. The wing planform is fixed.

Two structure design parameters control the *relative thickness change* of the wing front and rear spars in combination with the upper and lower sheet thicknesses of the

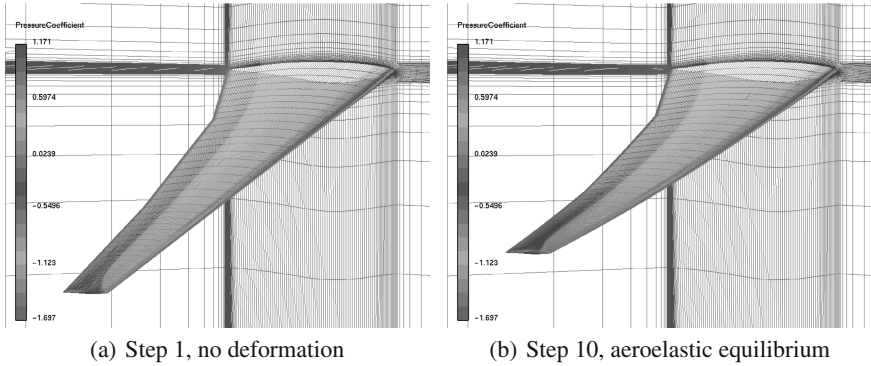


Fig. 11 Undeformed and deformed wing with pressure coefficient distribution and CFD mesh for the CFD/CSM coupled iterative process.

wing box. The stiffness and the weight of the wing are depending on these structure parameters.

Mesh deformation is a crucial component here. On the one hand, the change in geometry shape design through the parametric CAD model (CATIA V5) is treated by mesh deformation, on the other hand, the deformation of the wing structure depending on the aerodynamic forces (in addition to other forces such as fuel weight, engines, etc.) is also covered by applying mesh deformation. The CFD/CSM coupling is displayed in figure [11].

The individual components of the process chain were used in the parallel, in-memory FlowSimulator environment [11], so that a time-consuming and data intensive exchange of files was not required. Compared to the former methods, which used file exchange, significant time savings of around 50 percent have been obtained. This a major step forward in an industrial context together with the accuracy improvements and reductions of uncertainties.

In figure [12(a)] and [12(b)] the results of the optimization are presented. Shown is the original geometry in the aeroelastic equilibrium and the optimized geometry with a significantly different twist distribution and bending.

Figure [13] shows the convergence of the required thrust during the optimization process using a gradient free Downhill Simplex optimizer [8]. After around 80 design changes the optimum has been reached nearly.

6.3 Application to Complex Configuration

It was found that FSDeformation in the parallel FlowSimulator environment could be applied successfully to very complex, industrially relevant problems. An example for a coupled CFD/CSM application for an complete aircraft in high-lift configuration with deflected flaps and slats is shown in Fig. [14].

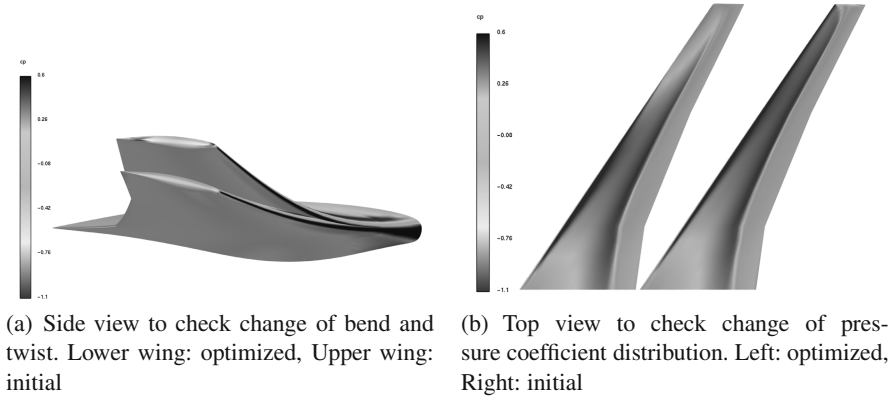


Fig. 12 Clean wing CFD-CSM optimization

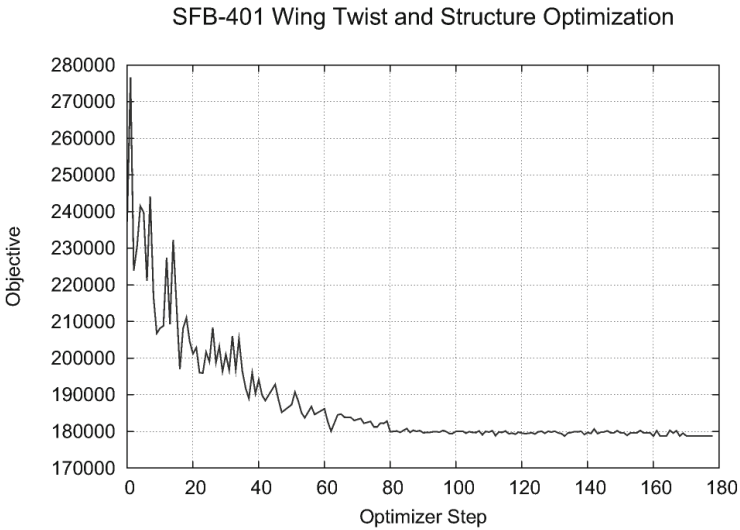


Fig. 13 Clean wing CFD-CSM optimization. Convergence of the required thrust in horizontal stationary flight during the optimization process.

7 Summary

This work has presented a mesh deformation module for the parallel simulation environment FlowSimulator. The module is based on the radial basis function interpolation approach.

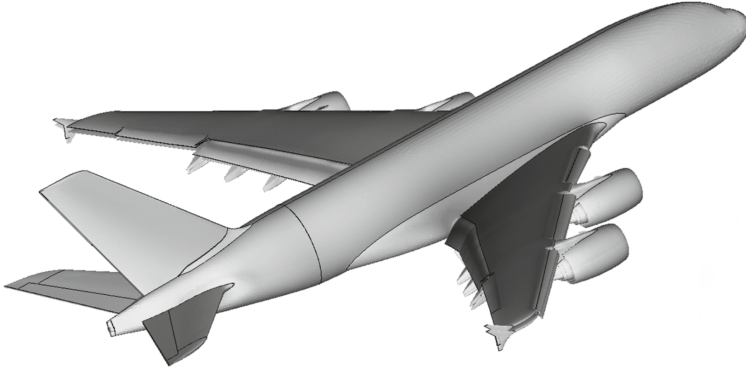


Fig. 14 Complex high-lift configuration for a CFD/CSM coupled simulation applying the new tool FSDeformation. Shown is the deformed and non-deformed geometry (see wing tip) due to a different aerodynamic load case.

The application fields of mesh deformation is manifold (unsteady simulations, shape optimization and aeroelasticity). In a next step, radial basis function interpolation and the mesh deformation module and its algorithm have been introduced. The module combines the radial basis function interpolation approach with a group-weighting and deformation-blending feature. This allows to move different surface groups/bodies independently from each other. Furthermore, the deformation blending provides an improved protection of boundary layer cells and allows the usage of unbounded radial basis functions and the extension of the radial basis function interpolation approach by a polynomial.

The group-weighting and deformation-blending uses the wall distance of the volume mesh nodes to the group boundaries. Hence a wall distance computation module has been implemented. It uses an advancing-front algorithm for the distance computation.

Additionally, the deformation module was extended with new deformation group features. These features allow to define deformation groups without creating base points and deformation vectors for this group. The features support far-field boundaries, symmetry planes or the rigid attachment of a boundary group to another boundary group.

Because the computational cost of the interpolation algorithm depends on the number of interpolation base points, the module offers four different methods for the reduction of the input base points. The first method uses an octree data-structure to select equidistant base points. The remaining three methods use in the first step this method as well. But in the second step they either use weighted distances for a modified equidistant reduction function, or they correct the interpolation error locally by selecting base points individually. The two weighted distance reduction methods use the difference to the mean deformation of the neighboring nodes or

the interpolation error of the base points not selected for the weight computation. All the different methods have been compared in terms of performance and interpolation error. Here the locally operating error correction method has shown very good results in performance/interpolation error efficiency. But the method is limited to locally supported radial basis functions. Because the radial basis function Euclid's Hat, which tends to infinity for an increasing input argument, has produced the lowest interpolation error, the interpolation error weighting method is the best advice.

The wall distance module and the deformation module have been parallelized with MPI. The theoretically perfect speedup of the interpolation method may only be affected by unbalanced node distributions over the parallel MPI processes.

The creation of the module FSDeformation by Airbus and its integration into the parallel FlowSimulator software environment has helped to reduce the uncertainties that occur in small geometry changes, which typically occur at aerodynamic shape design. Numerical shape optimization of components by using the improved grid deformation technique for unstructured grids has improved, reducing uncertainties related to mesh dependencies of the numerically obtained flow solutions.

Finally, applications for the deformation module in cooperation with the flow solver TAU and a structure module has been demonstrated. It has shown that the deformation module can play a key position in future computation chains using the simulation environment FlowSimulator with a perfect speedup for the mesh deformation method. Additionally, the example has illustrated that the coupling of different programs by using FlowSimulator can minimize intensive and time-consuming file input/output operations, due to the fact that both tools use the same main memory address space. The computational time and hence the cost of an optimization has been reduced considerably.

References

- [1] FlowSimulator: Common simulation environment for integrated parallel CFD applications, <http://dev.as.dlr.de/gf/project/fsdm>
- [2] Allen, C.B., Rendall, T.C.S.: Efficient mesh motion using radial basis functions with data reduction algorithms. In: AIAA Paper 2008-305, Department of Aerospace Engineering, University of Bristol, Bristol, Great Britain (January 2008)
- [3] Baxter, B.J.C.: The Interpolation Theory. PhD thesis, Trinity College. University of Cambridge (August 1992)
- [4] Beckert, A., Wendland, H.: Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. Technical report, Institute of Aeroelasticity, German Aerospace Center (DLR), Institute for Applied Mathematics, University of Göttingen (March 2000)
- [5] Buhmann, M.D.: Radial Basis Functions. Cambridge University Press (April 2003)
- [6] Heinrich, R., Kroll, N., Neumann, J., Nagel, B.: Fluid-Structure Coupling for Aerodynamic Analysis and Design - A DLR Perspective. In: 46th AIAA Aerospace Sciences Meeting and Exhibit 2008, Reno (2008)

- [7] Kroll, N., Schwamborn, D., Becker, K., Rieger, H., Thiele, F.: MEGADESIGN and MegaOpt - German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design: Results of the Closing Symposium of the MEGADESIGN and MegaOpt Projects. Springer (2007)
- [8] Nelder, J.A., Mead, R.: A simplex method for function minimization. *Computer Journal* 7, 308 (1965)
- [9] Ohtake, Y., Belyaev, A., Seidel, H.: Multi-scale and adaptive cs-rbfs for shape reconstruction from cloud of points (2003)
- [10] Stickan, B.: Implementation and extension of a mesh deformation module for the parallel FlowSimulator software environment. Diploma Thesis at Airbus Deutschland GmbH, Bremen and Chair for Computational Analysis of Technical Systems, RWTH Aachen (2009)
- [11] Tucker, P.G.: Differential equation-based wall distance computation for DES and RANS. PhD thesis, Department of Engineering, Fluid Dynamics Research Centre, University of Warwick (October 2002)
- [12] Wendland, H.: Fast evaluation of radial basis functions: Methods based on partition of unity. In: *Approximation Theory X: Wavelets, Splines, and Applications*, pp. 473–483. Vanderbilt University Press (2002)
- [13] Wigton, L.B.: *Optimizing cfd codes and algorithms for use on cray computers*, frontiers of computational fluid dynamics. World Scientific Publishing Co. Pte. (1998)

Part V
Stochastic Uncertainties

Statistical Analysis of Parameter Variations Using the Taguchi Method

A. Wolf, D. Henes, S. Bogdanski, Th. Lutz, and E. Krämer

Abstract. This project is part of the MUNA-project. It deals with the investigations of uncertainties of a CFD-Simulation caused by variation of numerical parameters, inflow conditions and geometrical parameters. To evaluate the influence of different parameters on the results of an experiment or simulation usually a full parameter analysis must be performed. A full parameter analysis requires a high number of simulations, rising with the number of parameters, which are investigated. In this study the Taguchi method, which is based on DoE (Design of Experiments) methods and well known for the optimization of production cycles, is applied. It reduces the number of required simulations dramatically and therefore also the costs. In this article the basics of the Taguchi Method are explained and a summary of the ANOVA-analysis, is given. The ANOVA analysis is used for the analysis of the results obtained by the Taguchi method. It delivers the relative influences of each investigated parameter and also the influence of interaction among different parameters can be obtained. The effect of the parameter variations on the CFD result is shown for two- and three dimensional simulations. Numerical, geometrical and also inflow conditions are investigated.

1 Introduction

The aim of this project is to investigate the uncertainties in CFD-results if different numerical (i.e numerical diffusion) and geometrical (i.e. trailing edge thickness) parameters are varied. Most of the settings used for CFD simulations are based on experience and it is unknown how the result behaves if these parameters are

A. Wolf · D. Henes · S. Bogdanski · Th. Lutz · E. Krämer
Institute of Aerodynamics and Gasdynamics, University of Stuttgart
e-mail: wolf,lutz@iag.uni-stuttgart.de,
Pfaffenwaldring 21, 70569 Stuttgart

changed. On the other hand there are also natural variations like fluctuating inflow conditions and production inaccuracies. Their effect is unknown as well. To evaluate the influence of all these parameters, usually a full parameter analysis must be performed. As such an analysis requires several thousand simulations, the Taguchi method is introduced in this article, which requires a much lower number of simulations. The Taguchi method enables to investigate the relative influence of different parameters on the CFD result by systematical variation of these parameters. It also enables to investigate the effect of the interactions between two parameters. It could be shown that in some cases the strongest influences on the CFD results stem from such interaction effects. In the first part of this article the Taguchi Method is explained and the results of the simulations are presented in the second part. The parameters are arranged in different groups depending whether they are numerical based, geometrical based or belong to inflow conditions. As test case the RAE2822 is chosen. The inflow conditions are taken from [1].

2 The Taguchi-Method

Subsequently the basic ideas of the Taguchi method are summarized. For more information it is referred to [2] to [6].

The Taguchi Method is based on the DoE-Method (Design of Experiments). DoE is a statistical method for quality management, usually used for the optimization of production cycles by determining the influence of chosen parameters on the quality of the final product. The aim of the DoE to is reduce the number of experiments needed to evaluate the influences of the parameters to a minimum in comparison to a full parameter variation. Especially the method developed by Gen'ichi Taguchi (born 1924) using orthogonal arrays for the experiment setup needs only a minimum of numbers of experiments [2]. The number of experiments required for a full parameter analysis rises exponential with the number of investigated parameters and the number of levels. If the influences of 4 parameters with 3 levels each should be investigated $3^4 = 81$ experiments would be necessary. In the case of 13 parameters with 3 levels 2197 experiments should be done. The Taguchi method needs for these cases only 9 and 27 experiments respectively. This results in a much shorter simulation time and less required resources. Another advantage of the Taguchi-Method is the possibility to evaluate also the effects of interactions between the parameters.

In the present investigations the Taguchi method is applied to assess the impact of different parameter settings on the results of CFD simulations. These results can be the calculated lift or the different drag portions (viscous drag and pressure drag). It enables to investigate several numerical and geometrical parameters in a very short time with an acceptable number of CFD analyses.

	A	B	C	D	
1	1	1	1	1	R1
2	1	2	2	2	R2
3	1	3	3	3	R3
4	2	1	2	3	R4
5	2	2	3	1	R5
6	2	3	1	2	R6
7	3	1	3	2	R7
8	3	2	1	3	R8
9	3	3	2	1	R9

Fig. 1 L_9 -Taguchi-Matrix

2.1 Orthogonal Arrays

The simulation procedure is described by orthogonal arrays called Taguchi matrices. Such an array is based on the number of variables and their interactions (columns) plus the number of levels which yield the number of lines. Usually a pre-defined matrix is chosen, but it is also possible to design a new matrix or to adapt an existing matrix. In Figure 1 a L_9 matrix is shown, which is suitable for the evaluation of the influence of 4 parameters with 3 levels each. The first column shows the number of the simulation. The following 4 columns (A-D) describe the settings for the 4 parameters. The results of the 9 simulations are recorded in the last column. The matrix describes the setting for each parameter. For example in simulation number 4 the parameter A is set to level 2, parameter B to level 1, parameter C to level 2 and parameter D is set to level 3. Several pre-defined matrices exist, shown in Figure 2 for different numbers of parameters and levels.

A modification of these matrices is possible. For example, the number of levels of a chosen parameter can be increased or reduced. But it has to be kept in mind that the sum of the degrees of freedom for all columns is not higher than the degree of freedom of the whole matrix. The degree of freedom of one column is equal to the number of levels minus 1 while the degree of freedom of the matrix is the number of simulations minus 1.

After all simulations have been done, the influence of each parameter is determined by an statistical ANOVA-Analysis (Analysis of Variances).

2.2 ANOVA-Analysis

With the ANOVA analysis (Analysis of Variances) the influence of each parameter and interaction on the result of the simulation is obtained. In this section a short summary of this statistical method is given. Thereby the variable y is the result of

Orthogonal Array	Number of lines	Max. No. of parameters	Max. Number of columns for given Number of levels			
			2	3	4	5
L ₄	4	3	3	-	-	-
L ₈	8	7	7	-	-	-
L ₉	9	4	-	4	-	-
L ₁₂	12	11	11	-	-	-
L ₁₆	16	15	15	-	-	-
L' ₁₆	16	5	-	-	5	-
L ₁₈	18	8	1	7	-	-
L ₂₅	25	6	-	-	-	6
L ₂₇	27	13	-	13	-	-
L ₃₂	32	31	31	-	-	-
L' ₃₂	32	10	1	-	9	-
L ₃₆	36	23	11	12	-	-
L' ₃₆	36	16	3	13	-	-
L ₅₀	50	12	1	-	-	11
L ₅₄	54	26	1	25	-	-
L ₆₄	64	63	63	-	-	-
L' ₆₄	64	21	-	-	21	-
L ₈₁	81	40	-	40	-	-

Fig. 2 Overview about pre-defined Taguchi matrices

the CFD simulation, n is the number of simulations and f is the degree of freedom. First of all the mean square error is determined by

$$SQ_m = \frac{(\sum y_i)^2}{n} = CF . \tag{1}$$

The degree of freedom of the mean square values is

$$f_{SQ_m} = n - (n - 1) \tag{2}$$

In the next step the total error square sum is determined.

$$SQ_{total} = (\sum y_i)^2 - CF . \tag{3}$$

It has the degree of freedom

$$f_{SQ_{total}} = n - f_{SQ_m} . \tag{4}$$

In the next equation the mean square error of the variance is calculated

$$SQ_A = \frac{(\sum y_{A1})^2}{n_{A1}} + \frac{(\sum y_{A2})^2}{n_{A2}} + \frac{(\sum y_{A3})^2}{n_{A3}} - CF , \tag{5}$$

exemplarily shown for a parameter A with 3 levels. The results of the simulations with the same level for parameter A are summed, squared and divided by the number

of simulations with the same level of parameter A. Then the sum of the mean square values CF is subtracted. The degree of freedom is

$$fSQ_A = n_{Levels} - fSQ_m, \tag{6}$$

where n_{Levels} are the number of levels of parameter A. The squared variances of the interactions are calculated by

$$SQ_{AxB} = \frac{(\sum y_{(AxB)i})^2}{n_{(AxB)i}} - CF - SQ_A - SQ_B, \tag{7}$$

shown exemplarily for parameter A and B. The results of the simulations where the level combination of parameter A and B is the same are summed, squared and divided by the number of results with the same level combination. Then the sum of the mean square values and the mean square errors of parameter A and B are subtracted.

In the next step the estimation variances of each parameter are estimated by

$$V_A = \frac{SQ_A}{f_A}, \tag{8}$$

again shown for parameter A. Then the evaluation of the critical F_i can be performed.

$$F_A = \frac{V_A}{V_F}. \tag{9}$$

This value shown here for the parameter A is compared to the so called Fisher values. The Fisher value is a reference value used for the determination of the differences of variances with a certain probability of confidence. Reference values are given in so called Fisher tables depending on the probability of confidence. If the calculated Fisher value is larger than the value given by the Fisher table, the null hypothesis will be rejected. It is assumed that the differences from the total mean value are not random (null hypothesis). Therefore the alternative hypothesis is valid. If the null hypothesis is rejected, it can be assumed that an influence of the tested parameter is existent. For the percental influence of each parameter the adjusted error square sum must be determined first.

$$SQ'_A = SQ_A - f_A V_F \tag{10}$$

Dividing this value by the squared variances and multiplying it with 100. the percentage influence of the parameter (here shown for parameter A) on the results of the simulation is obtained.

$$p_A = \frac{SQ'_A}{SQ_A} \cdot 100 \tag{11}$$

2.3 Parameter Interaction

The influence of possible interactions between two parameters can be captured by the Taguchi Method as well. For each interaction one (for parameters with 2 levels) or two (for parameters with more than 2 levels) columns must be reserved in the orthogonal array. This increases the number of required columns and also the number of simulations. Thus the advantage of requiring less simulations than a standard parameter variation is reduced. But on the other hand, the influence of interactions between the investigated parameters can be evaluated. This won't be possible if a normal parameter simulation is used. For several pre-defined Taguchi matrices so-called interaction-matrices are published. They describe which columns are reserved for which parameter interaction. The interaction-matrix for the L_8 Matrix is shown in Figure 3. For example column 3 must be reserved to examine the interaction between the parameters number 1 and 2. The parameters originally stored in these columns would move to column 4 if no other interactions are investigated.

2.4 Error Determination

For each analysis the error can be calculated. During the investigations carried out at the IAG it was recognized that this error is very large if one or more simulations did not converge. This is obvious since the deviation caused by the non-convergence is usually larger than the deviation caused by the small parameter variations of the Taguchi analysis. Thus the convergence of every simulation is crucial and absolutely necessary for a reliable quantification of the parameter influences and interactions. The error that occurred can be evaluated similar to the influences in the ANOVA analysis.

No. of column	1	2	3	4	5	6	7
1	-	3	2	5	4	7	6
2		-	1	6	7	4	5
3			-	7	6	5	4
4				-	1	2	3
5					-	3	2
6						-	1
7							-

Fig. 3 L_8 -Interaction-Matrix

$$SQ_F = SQ_{total} - \sum_{i=1}^{n_{parameters}} SQ_i - \sum_{j=1}^{n_{interactions}} SQ_j \quad (12)$$

The degree of freedom of the error term is calculated as follows,

$$f_F = f_{total} - \sum_{i=1}^{n_{parameters}} f_i - \sum_{j=1}^{n_{interactions}} f_j \quad (13)$$

If the degree of freedom is non-zero, the error variance can be determined.

$$V_F = \frac{SQ_F}{f_F}, \quad (14)$$

The degree of freedom becomes zero if all columns of the Taguchi matrix are occupied with parameters. In this case an error determination as shown above is not possible. Instead the so-called pooling-method must be used. This method takes the parameters with the smallest influences to determine an error term. Following equation describes this method exemplarily for two parameters C and D.

$$V_{F2} = \frac{SQ_{C+D}}{f_{C+D}}, \quad (15)$$

In [2] it is mentioned that at least half of the degrees of freedom f_{total} should be used for this method to determine the error of the investigation. In the present investigation all runs except run 1 are performed with Taguchi matrices which are not fully packed with parameters. Therefore usually the first error determination method is used.

3 Simulation Setup

To investigate the influence of different numerical parameters, steady and unsteady RANS simulations are performed. The TAU code developed by the DLR is used for all simulations. Test object is the RAE 2822 airfoil. The inflow conditions are chosen according to the known Case 9 [1].

$$Ma_\infty = 0.73, T = 288.15K, Re = 6.5e+6, \alpha = 2.80^\circ$$

The following section gives a summary of the performed investigations and their results. As mentioned above the Taguchi method allows to evaluate also the influence of the interactions between the parameters. Therefore, the different parameters are partitioned into several groups. Each group contained physically or numerically related parameters.

For the steady simulations (Run 1 till 5) the influences of each particular parameter on the results for the lift coefficient C_l , total drag coefficient C_d , pressure drag coefficient C_{dp} and viscous drag coefficient C_{dv} are investigated.

In case of the unsteady simulation the influence on the averaged lift and drag coefficients C_{lmean} and C_{dmean} , their amplitudes C_{lA} and C_{dA} as well as their RMS values C_{lrms} and C_{drms} is shown. Additionally, the influence on the buffet frequency is described.

All simulations are performed with the TAU-version 2008.1.1 and the meshes are all generated with GridGEN 15.11. The mesh generation is script based and all meshes are structured (C-mesh). For the mesh variation (Run 1 and 3) several different meshes are generated, each with another resolution, y^+ value and grid resolution at leading and trailing edge. The mesh resolution is based on the number of points along the airfoil surface (upper and lower side).

No. of grid points:

19580 with 128 grid points on the airfoil surface,

79476 with 256 grid points on the airfoil surface,

320228 with 512 grid points on the airfoil surface,

Figure 4 shows the leading edge region of the finest mesh with 512 points along the airfoil surface. The TAU settings are chosen as follows:

k2-dissipation factor: 0.5

k4-dissipation factor: 64

CFL number: 1.5

Turbulence Model: Menter-SST

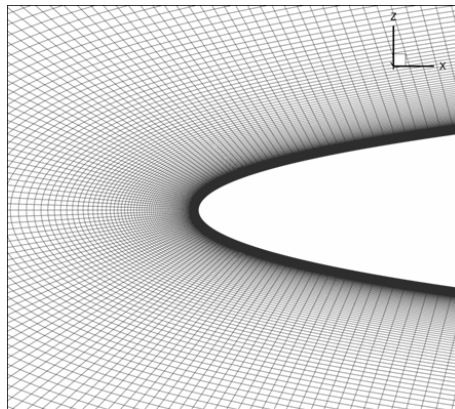


Fig. 4 Leading edge region of the finest mesh

4 Results

Run 1: Mesh resolution, Boundary layer resolution

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: y^+ , Number of grid points on the airfoil surface, mesh resolution leading edge, mesh resolution trailing edge

In the first case the influences of different grid parameters, including the y^+ value, number of grid points on the airfoil surface, which determines the overall mesh resolution, as well as the local grid resolution at leading (LE) and trailing edge (TE) are evaluated. The last two mentioned properties are defined by the distance of the points on the airfoil surface. The mesh resolution at leading and trailing edge is set to coarse, medium and fine. The total value for each level depends on the total number of grid points on the surface. Defining the resolution at leading and trailing edge this way, guarantees meshes which are similar to each other. Table 1 gives an overview of the resulting total values, given in per mille of the chord length. The y^+ value determines the height of the first cell row at the airfoil surface. The values for the three levels are 0.5, 1 and 2. Figure 5 shows the influences of the varied parameters on the coefficients for drag and lift given in percent. The influences of the interactions are shown as well. p_{AC} for examples denotes the influence of the interaction between parameters A and C. The error occurred during the ANOVA-analysis is given by p_F . This value does not represent the influence of the error on the CFD result. It is just an indication for the certainty of the evaluated parameter influences. In this study only interactions with the y^+ parameter are investigated. The reason is that four parameters are examined at once, and that there is no suitable Taguchi Matrix which includes four parameters and all their interactions. In Figure 5 it is shown that the influence of the number of grid points on the airfoils surface prevails in comparison to the other parameters. Especially for the results of lift, total drag and pressure drag. In case of the viscous drag also the y^+ value shows some significant impact. This is due to the fact that this parameter effects the resolution of the boundary layer. The mesh resolution at trailing and leading edge shows only a small contribution. No interaction between the y^+ value and the other parameters is detected.

Table 1 Total value of the grid point distance at leading and trailing edge for different mesh resolutions, given in per mille

Level	512	256	128
1-rough	1	2	4
2-middle	0.5	1	2
3-fine	0.25	0.5	1

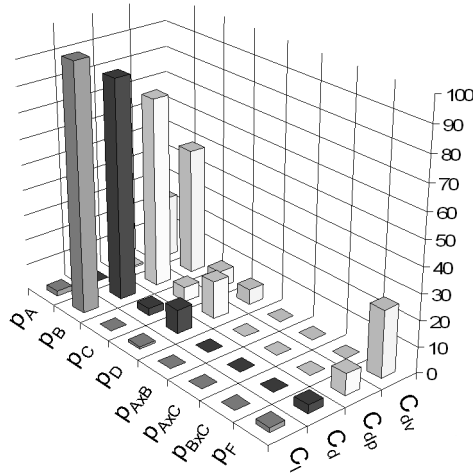


Fig. 5 Results of run 1. A = y^+ , B = number of grid points on airfoil’s surface, C = mesh resolution TE, D = mesh resolution LE

Run 2: Inflow conditions

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: Reynolds number, Mach number, angle of attack

For the investigation on the inflow conditions the Reynolds number is varied by $\pm 3 \cdot 10^5$, the Mach number by ± 0.001 and the angle of attack by $\pm 0.01^\circ$. In this study the fine mesh with 512 grid points on the surface of the airfoil is chosen. The ANOVA analysis yields a large impact of the Reynolds number and the angle of attack on the lift coefficient C_l as it can be seen in Figure 6. This result is comprehensible especially for the angle of attack. But it is also known that the lift increases with a rising Reynolds number. In transonic region the Mach number has a large influence on the total drag values because the pressure drag which is also mainly influenced by the Mach number has a large share in the total drag. The viscous drag, however, is only influenced by the Reynolds number. But it has only a low impact on the total drag.

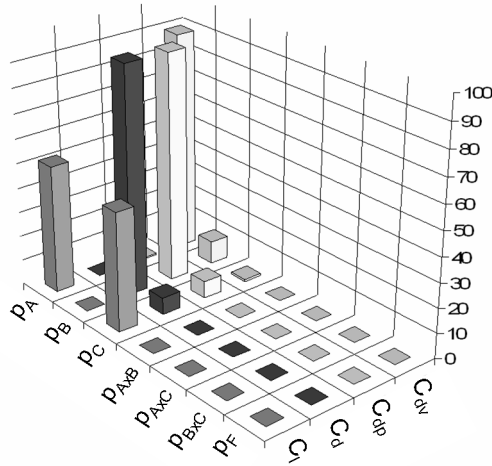


Fig. 6 Results of run 2. A = Reynolds number, B = Mach number, C = angle of attack

Run 3: Turbulence modeling (a)

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: Mesh resolution, CFL-number, numerical diffusion

In this case the mesh resolution at leading and trailing edge is set to the medium value. The levels of the CFL number are set to 1.5, 3.25 and 5. The numerical diffusion terms are changed simultaneously. Their levels are (1/32; 1.00), (1/64; 0.50) and (1/128; 0.25) for k2 and k4 respectively. This study is done for several turbulence models like Menter-SST, Spalart Allmaras with Edwards correction (SAE), Linear Algebraic Stress Model (LEA) and for the Reynolds Stress Model (RSM). The results are very similar for all turbulence models. Therefore, only the result for the SAE model is exemplified in Figure 7. The analysis shows that the mesh resolution has a main impact on all results again. The CFL number has no influence. This behaviour agrees well with the theory as long as all simulations are converged like in the present case. The influence of the numerical diffusion is also very small, but the interaction between the mesh resolution and the numerical diffusion shows a larger influence on the results. This is reasonable as the mesh has also a damping effect on the solution if it is too coarse. Since in this study only three parameters are investigated, the evaluation of all interaction effects is possible by using a L_{27} Matrix.

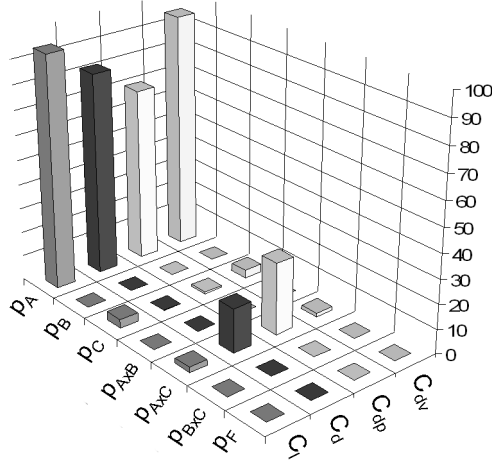


Fig. 7 Results of run 3. A = number of points on airfoil’s surface, B = CFL-number, C = Diffusion terms

Run 4: Turbulence modeling (b)

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: Turbulence model, CFL-number, numerical diffusion

This is based on the results of run 3, which was performed for several turbulence models (Menter-SST, SAE, LEA). By rearranging the performed simulations, it is possible to replace the mesh resolution parameter by the turbulence model. A new ANOVA analysis shows that in this case the influence of the turbulence model out-balances all other parameters. This is not surprising because it is known that the turbulence model has a large impact on the CFD result, and the other parameters are not varied to such an extent to compete with the turbulence model.

Run 5: Geometry

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: Trailing edge thickness, number of geometry-defining points, bump height

The three levels of the trailing edge thickness are 0%, 0.5% and 1% of the chord length. The number of points defining the contour of the airfoil are set to 100, 200 and 400. Finally a sinus function is superposed with the leading edge contour to

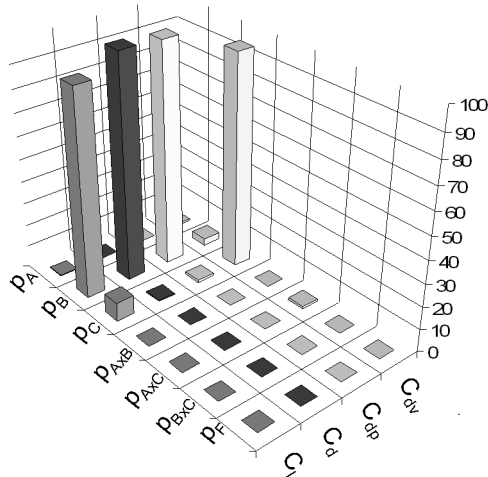


Fig. 8 Results of run 5. A = number of geometry-defining points, B = Trailing edge thickness, C = Bump height.

simulate bump-shaped disturbances in the leading edge region. These bumps represent the uncertainty in fabrication process. The amplitude levels of these bumps are set to $1 \cdot 10^{-4}$, $2 \cdot 10^{-4}$ and $3 \cdot 10^{-4}$ times the chord length. For the simulation the mesh with 256 points on the airfoil’s surface is chosen. The trailing edge is discretized with 56 cells and 110 cells for a trailing edge height of 0.5% and 1% chord. The results show that the number of points has no influence. This is due to the fact that even 100 points are still sufficient to describe the airfoil’s contour nicely. The coefficients for lift, total drag and pressure drag are mainly affected by the trailing edge thickness while the viscous drag is influenced by the bump height. The bump height has an impact on the pressure distribution along the airfoil’s surface. The acceleration and deceleration along the wavy surface changes the load on the boundary layer which affects the viscous drag.

Run 6: Unsteady simulation

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: CFL-number, Number of inner iterations, Δt

To get unsteady flow characteristics, the angle of attack is increased to 5° . For this angle of incidence buffet effects occur, i. e. that the shock on the upper surface starts to move with a distinct frequency. Therefore also the force coefficients fluctuate. In case of an unsteady simulation not only the mean force coefficients but also their

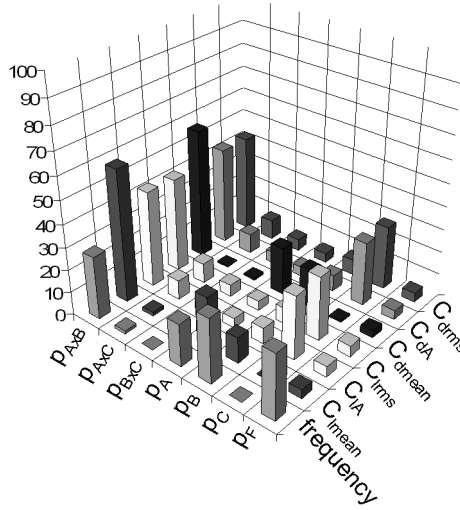


Fig. 9 Results of run 6. A = unsteady physical time step size; B = Number of inner iteration; C = CFL-number.

amplitudes, rms-values and frequencies can be analyzed. The CFL number is set to 3.4, 1.7 and 0.8 for the coarse multigrid and to 10, 5 and 2.5 for the fine multigrid. The ratio t_{char} to Δt is defined to 20, 40 and 80. Where $t_{char} = U_{\infty}/c$ is the time the air needs to flow over the airfoil. The levels for the number of inner iterations are 50, 100 and 200. Looking at the results plotted in Figure 9 it must be said first that the CFL number has a larger impact on several results. As mentioned before this should not be the case if all simulations are converged. But in this study the inner iterations of some unsteady simulations are not converged. This explains the recognized influence of the CFL number. Furthermore, the unsteady physical time step size $\Delta t/t_{char}$ has an influence on the mean values of the force coefficients and the frequency but not on the unsteady parts of the force coefficients, the rms-value and the amplitude. The unsteady physical time step size has also an indirect effect on all parameters through the interaction with the number of iterations. This interaction often has the strongest influence of all parameters. All other interaction effects can be neglected. The number of inner iterations has a direct influence on the frequency and the mean values.

Run 7: Robustness test

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: Reynolds number, Mach number, angle of attack

The study for the inflow conditions is repeated, but the variation of the parameters is increased to three times the value as in run 2. The Mach number is now varied by ± 0.003 , the angle of attack by $\pm 0.03^\circ$ and the Reynolds number is varied by $\pm 9 \cdot 10^5$. This study provides almost exactly the same results as run 2. This confirms the robustness of the Taguchi Method that shows that the obtained results are also valid for larger (or smaller) parameter variations.

Run 8: 3D Simulation

Taguchi Matrix: L_{27}

No. of levels: 3 for each parameter

Parameters: Reynolds number, Mach number, angle of attack

Finally the Taguchi Method is used for the analysis of 3D simulations of the generic SFB-401 wing. Geometry and mesh are provided by the DLR. The mesh has 9.5 million points. The boundary layer is resolved with 80 layers in wall normal direction. The boundary conditions are chosen to $Ma = 0.8$, $Re = 23.5 \cdot 10^6$ and angle of attack of 1° . For the 3D simulation a Taguchi analysis of these boundary conditions is performed as it is already done for the RAE 2822 2D-airfoil shown before (Run 2). The variation of the Mach number is 0.001 and the angle of attack is varied by 0.01° . These variations are the same like in the investigation of the 2D-airfoil since the total values of these parameters are similar for the 3D and the 2D case. The Reynolds number is varied stronger than before as the Reynolds number of the 3D SFB-401 case is much higher than for the 2D test case. Therefore, it is varied by $\pm 1 \cdot 10^6$ instead of $\pm 3 \cdot 10^5$. For the simulations the SAO turbulence model is used. The dissipation factors are set to 0.5 and 64 for k2 and k4 respectively. Figure 10 shows a cut through the mesh at $\eta=0.77$.

The c_p distribution is shown in Figure 11. Inboard a shock appears due to supersonic flow at $x/c=0.6$ for $\eta=0.23$. Further outboard at $\eta=0.77$ no shock is observed. These results are extracted from the simulation with $Ma=0.799$, $Re=22.5 \cdot 10^6$ and angle of attack $\alpha=1.01^\circ$. The ANOVA analysis yielded results comparable to the 2D case but shows up also some differences (see Figure 12). The viscous drag is again mainly affected by the Reynolds number. Mach number and angle of attack have no influence. Looking on the pressure drag it can be recognized that the Mach number has the largest influence followed by the angle of attack which has some small contribution. The influence of the angle of attack is increased compared to the 2D airfoil as it has some impact on the induced drag which is a pressure drag as well and occurs only for lift producing 3D geometries. The results for the total drag show some differences to the 2D case. The impact of the Reynolds number is much higher for the 3D case. This can be explained by the larger contribution of the viscous drag to the total drag compared to the 2D case. For the 3D case the viscous drag is of the same size as the pressure drag while for the 2D case the pressure drag was twice the viscous drag. Therefore, the Reynolds number has a larger impact on the total drag for the 3D case. However, the influence of the Reynolds number on the

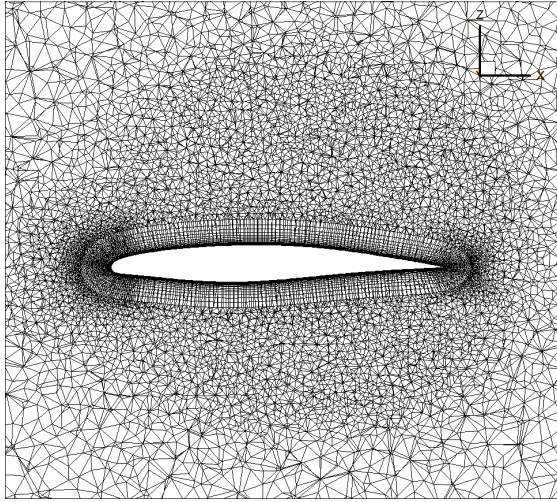


Fig. 10 Mesh of the SFB-401 wing. Cut at $\eta=0.77$

lift coefficient is smaller than in case of the 2D airfoil. This can be explained by the larger Reynolds number which is four times higher than for the 2D case. Therefore, the influence of the viscous forces on the lift will be smaller than for the 2D case. The higher contribution of the viscous drag to the total drag, mentioned before, is rather explained by a different geometry than by a different Reynolds number.

5 TAU Implementation

To automatize the procedure and to reduce the required effort it was decided to implement the Taguchi method into the TAU Code by using Python scripts. The new script, which was published with the TAU 2012 release supports the user during the selection of the Taguchi Matrix, its modification and in the consideration of interactions. Several parameters can be chosen from a given list, which can be extended at any time. After defining parameters and their levels para files (TAU control file), job files (to start the simulation on the high performance computer) and required folders are setup automatically for all required simulations based on the informations from the chosen Taguchi-Matrix. The simulations can be either started in serial mode as one "big" job or in parallel mode. The second approach needs to be used if the parameters to be investigated include changes in the meshes or turbulence models. Both can not be changed during a running TAU job. To support the user in this case a text file is written out to give information which mesh or turbulence model needs to be used in which simulation. When all simulations have been successfully finished the results can be examined using the ANOVA analysis. This step is automatized by a script as well. The script takes the Taguchi Matrix,

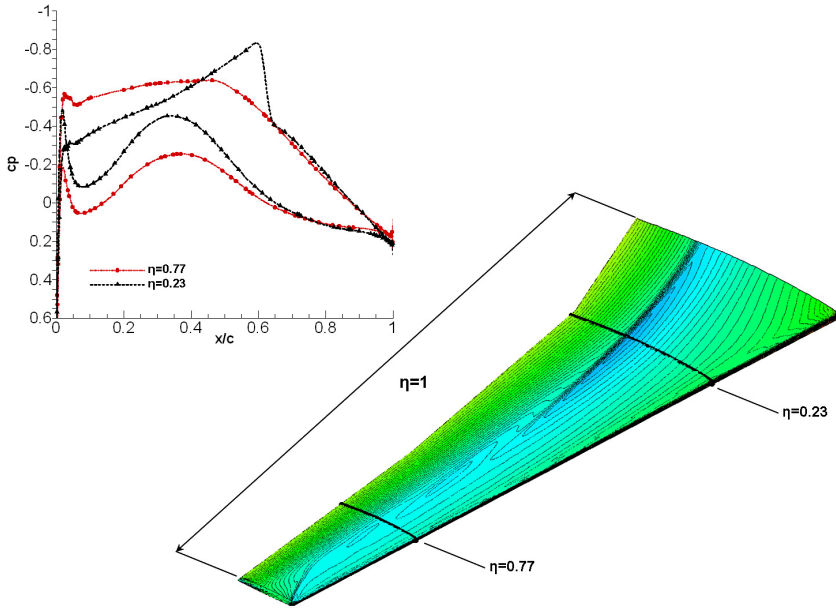


Fig. 11 Cp distribution of the SFB-401 wing. Cut at $\eta=0.77$ and 0.23

searches for the required TAU output files and reads in the results defined by the user before. Finally, the results of the ANOVA analysis are stored in a solution file, which can be easily read by Excel or other visualization tools. Validation of the program showed perfect agreement with the manual analysis and the required time is massively reduced as the hole setup for the TAU simulations is automatized. Still some knowledge about the Taguchi-Method is required. Therefore, a detailed users guide is added to the scripts including a detailed description of the Taguchi-Method and several examples.

6 Summary and Conclusion

In the described studies CFD settings are systematically varied by using a minimum number of simulations to quantify the relative impact of several numerical and geometrical parameters on the CFD result. The Taguchi method is applied to reduce the number of simulations in comparison to a full parameter variation. Statistical analysis (ANOVA) gave informations about the relative parameter influence on the CFD results like lift and drag. It also gives information on interactions between different parameters and their influence on the result. During the studies steady and unsteady test cases are performed and in both cases the results reflect long lasting experiences in CFD simulation. It is also shown that the results achieved by the

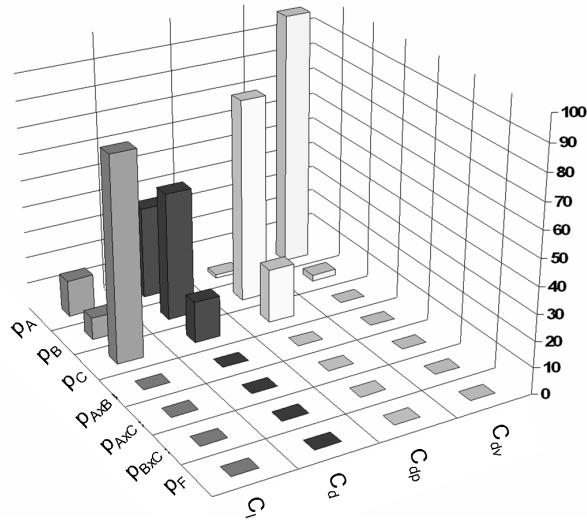


Fig. 12 Results of run 8. A = Reynolds number, B = Mach number, C = angle of attack

Taguchi method are robust to stronger variations of all parameters. This makes the results more generally valid, at least as long as the parameters are not varied too strongly. Therefore, the performed study shows that the Taguchi method is a suitable systematic method to check the accuracy of CFD simulations. To reduce the effort even more the Taguchi-Method was implemented into the TAU code via Python scripts. The scripts support the user during the selection of the Taguchi matrix and setup automatically all necessary TAU input files and folders.

References

- [1] AR-138, Experimental Data Base for Computer Program Assessment, ch. A6-1 (1979)
- [2] Klein, B.: Versuchspanung-DOE: Einführung in die Taguchi/Shainin-Methodik, 2nd edn., Oldenburg
- [3] Bläsing J.P. (Hrsg.): Workbook DoE, Design of Experiments nach G. Taguchi, TQU
- [4] Roos, P.J.: Taguchi Techniques for Quality Engineering. 2. McGraw-Hill, Auflage
- [5] Hering, Triemel, Blank: Qualitätssicherung für Ingenieure. VDI Verlag
- [6] Lutz, T., Sommerer, A., Wagner, S.: Parallel Numerical Optimisation of Adaptive Transonic Airfoils. In: Fluid Mechanics and its Applications, vol. 73. Kluwer Academic Publishers

Numerical Methods for Uncertainty Quantification and Bayesian Update in Aerodynamics

Alexander Litvinenko and Hermann G. Matthies

Abstract. In this work we research the propagation of uncertainties in parameters and airfoil geometry to the solution. Typical examples of uncertain parameters are the angle of attack and the Mach number. The discretisation techniques which we used here are the Karhunen-Loève and the polynomial chaos expansions. To integrate high-dimensional integrals in probabilistic space we used Monte Carlo simulations and collocation methods on sparse grids. To reduce storage requirement and computing time, we demonstrate an algorithm for data compression, based on a low-rank approximation of realisations of random fields. This low-rank approximation allows us an efficient postprocessing (e.g. computation of the mean value, variance, etc) with a linear complexity and with drastically reduced memory requirements. Finally, we demonstrate how to compute the Bayesian update for updating a priori probability density function of uncertain parameters. The Bayesian update is also used for incorporation of measurements into the model.

1 Introduction

Nowadays, the trend of numerical mathematics is often trying to resolve inexact mathematical models by very exact deterministic numerical methods. The reason of this inexactness is that almost each mathematical model of a real world situation contains uncertainties in the coefficients, right-hand side, boundary conditions, initial data as well as in the computational geometry. All these uncertainties can affect the solution dramatically, which is, in its turn, also uncertain. The information of the interest is usually not the whole set of realisations of the solutions (too much data), but some other stochastic information: cumulative distribution function, probability density function, mean value, variance, quantiles, exceedance probability etc.

Alexander Litvinenko · Hermann G. Matthies
Institute of Scientific Computing, Technische Universität Braunschweig,
Hans-Sommer str. 65, 38106, Braunschweig, Germany
e-mail: wire@tu-bs.de
<http://www.wire.tu-bs.de>

During the last few years, one can see an increasing interest in numerical methods for solving stochastic computational fluid dynamic (CFD) problems [3, 8, 17, 19, 24, 26]. In this work we consider an example from aerodynamics, described by a system of Navier-Stokes equations with a k- ω turbulence model. Uncertainties in parameters such as the angle of attack α and Mach number are modelled by random variables, uncertainties in the shape of the airfoil are modelled by a random field [14, 12]. Uncertain output fields such as pressure, density, velocity, turbulence kinetic energy are modelled by random fields as well. The lift, drag and moments will be random variables.

We assume that there is a solver which is able to solve the deterministic (without uncertainties) Navier-Stokes problem. In this work we used the TAU code (developed in DLR) with k- ω turbulence model [4]. We also assume that spatial discretisation of the airfoil is given. Our job is the appropriate modelling of uncertainties and developing stochastic/statistical numerical techniques for further quantification of uncertainties. At the same time, due to the high complexity of the deterministic solver, we are allowed to use only non-intrusive stochastic methods such as Monte Carlo or collocation methods. So, we are interested in methods which do not require changes in the deterministic code.

The rest of the paper is structured as follows. In Section 3 we describe the problem and discretisation techniques, such as the Karhunen-Loève expansion (KLE) [16] and polynomial chaos expansion (PCE) of Wiener [25]. In Section 2.1 we explain how we model uncertainties in the parameters angle of attack and Mach number. Uncertainty in the airfoil geometry is described in Section 2.2. The low-rank response surface is presented in Section 4. To avoid large memory requirements and to reduce computing time, low-rank techniques for representation of input and output data (solution) were developed in Section 5. Section 7 is devoted to the numerical results, where we demonstrate the influence of uncertainties in the angle of attack α , in the Mach number Ma and in the airfoil geometry on the solution - drag, lift, pressure and absolute friction coefficients. The strongly reduced memory requirement for storage stochastic realisations of the solution is demonstrated as well. In Section 6 we demonstrate how to use the Bayesian update (BU) for improving the statistical description of the random airfoil geometry. Section 7 is devoted to other numerical experiments.

2 Statistical Modelling of Uncertainties

The problem to consider in this work is the stationary system of Navier-Stokes equations with uncertain coefficients and parameters:

$$\begin{aligned} v(x, \omega) \cdot \nabla v(x, \omega) - \frac{1}{Re} \nabla^2 v(x, \omega) + \nabla p(x, \omega) &= g(x) \quad x \in \mathcal{G}, \quad \omega \in \Omega \\ \nabla \cdot v(x, \omega) &= 0 \end{aligned} \quad (1)$$

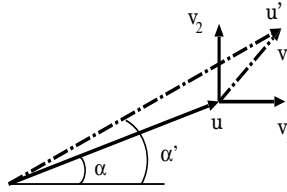


Fig. 1 Two random vectors \mathbf{v}_1 and \mathbf{v}_2 model free-stream turbulence, \mathbf{u} and \mathbf{u}' old and new free stream velocities, α and α' old and new angles of attack

with some initial and boundary conditions. Here v is velocity, p pressure and g the right-hand side, the computational domain \mathcal{G} is RAE-2822 airfoil with some area around. Examples of uncertain parameters are the angle of attack α and the Mach number Ma . Uncertainty in the airfoil geometry is modelled via random field (see Section 7).

2.1 Modelling of Uncertainties in Parameters

We assume that the free-stream turbulence in the atmosphere randomly and simultaneously changes the velocity vector or, what is equivalent the Mach number Eq. 4 and the angle of attack Eq. 3. One should not mix this kind of turbulence with the turbulence in the boundary layer reasoned by friction. It is assumed that turbulence vortices in the atmosphere are comparable with the size of the airplane. The free-stream turbulence in the atmosphere is modelled by two additionally axes-parallel velocity vectors $\mathbf{v}_1 := \mathbf{v}_1(\theta_1)$ and $\mathbf{v}_2 := \mathbf{v}_2(\theta_2)$ (Fig. 1), which have Gaussian distribution [13]. We model the free-stream turbulence via two random vectors (in 3D it will be three vectors) \mathbf{v}_1 and \mathbf{v}_2 which change α and Ma (see Fig. 1):

$$\mathbf{v}_1 = \frac{\sigma \theta_1}{\sqrt{2}} \quad \text{and} \quad \mathbf{v}_2 = \frac{\sigma \theta_2}{\sqrt{2}},$$

where θ_1 and θ_2 are two Gaussian random variables with zero mean and unit variance, $\sigma := \mathbb{I} u_\infty$, \mathbb{I} the mean turbulence intensity and u_∞ the undisturbed free stream velocity beyond the boundary layer. This mean turbulence intensity is often used for characterising turbulence in a wind tunnel. For instance, $\mathbb{I} = 0.001$ means low turbulence, $\mathbb{I} = 0.002$ middle and $\mathbb{I} = 0.005$ high turbulence.

Denoting

$$\theta := \sqrt{\theta_1^2 + \theta_2^2}, \quad \mathbf{v} := \sqrt{\mathbf{v}_1^2 + \mathbf{v}_2^2}, \quad \beta := \arctg \frac{\mathbf{v}_2}{\mathbf{v}_1} \quad \text{and} \quad z := \frac{\mathbb{I} \theta}{\sqrt{2}}, \quad (2)$$

and performing easy geometrical computations, obtain the new angle of attack and the new Mach number:

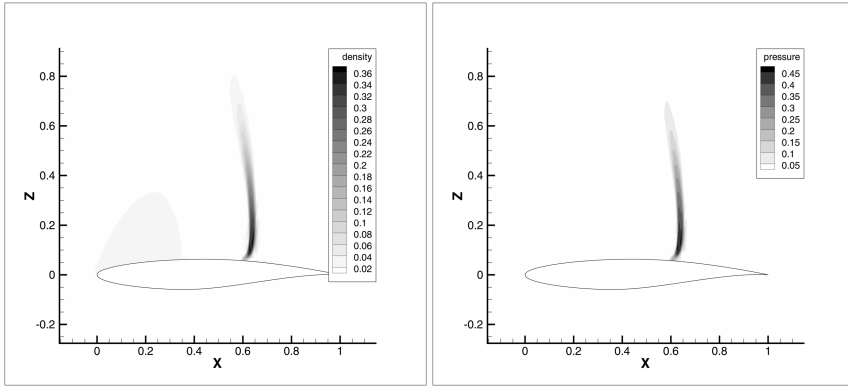


Fig. 2 (left) The difference $\Delta\rho := |\rho - \bar{\rho}|$ between the deterministic density $\rho := \rho(\alpha, Ma)$ and the mean density $\bar{\rho}$. (right) The same is for the pressure $\Delta p := |p - \bar{p}|$. Here $\bar{\rho} \in (0.5, 1.2)$ and $\bar{p} \in (0.7, 1.3)$.

$$\alpha'(\theta_1, \theta_2) = \arctg \frac{\sin \alpha + z \sin \beta}{\cos \alpha - z \cos \beta}, \tag{3}$$

$$Ma'(\theta_1, \theta_2) = Ma \sqrt{1 + \frac{\mathbb{I}^2 \theta^2}{2} - \sqrt{2} \mathbb{I} \theta \cos(\beta + \alpha)}. \tag{4}$$

Further we study how uncertainties in α and Ma spread into the solution. We note that uncertainties in α and in Ma can be modelled in a different way (see e.g. [23], [27]). From the construction one can see that $\overline{Ma'}$:= $\mathbb{E}(Ma'(\theta_1, \theta_2))$ and $\overline{\alpha'}$:= $\mathbb{E}(\alpha'(\theta_1, \theta_2))$ are equal to the deterministic values Ma and α , here $\mathbb{E}(\cdot)$ is the mathematical expectation. In Fig. 2 (left) we compare the deterministic density $\rho(\alpha, Ma)$ with the $\bar{\rho} := \mathbb{E}(\rho(\alpha'(\theta_1, \theta_2), Ma'(\theta_1, \theta_2)))$ for the Case 9 ($\alpha = 2.79$, $Ma := 0.734$). In Fig. 2 (right) we do the same comparison for the deterministic pressure. One can see a large difference in the shock position. This large difference motivates us to model uncertainty in α and in Ma .

2.2 Modelling of Uncertainties in the Airfoil Geometry

We model uncertainties in the geometry of RAE-2822 airfoil via random boundary perturbations:

$$\partial\mathcal{G}_\varepsilon(\omega) = \{x + \varepsilon \kappa(x, \omega)n(x) : x \in \partial\mathcal{G}\}, \tag{5}$$

where $n(x)$ is the normal vector in a point x , $\kappa(x, \omega)$ a random field, \mathcal{G} the computational geometry and $\varepsilon \ll 1$. We assume that the covariance function is of Gaussian type

$$\text{cov}_\kappa(\kappa_1, \kappa_2) = \sigma^2 \cdot \exp(-d^2), \quad d = \sqrt{|x_1 - x_2|^2/l_1^2 + |z_1 - z_2|^2/l_2^2},$$

Table 1 Statistics obtained for uncertainties in the airfoil geometry. We used the Gaussian covariance function, PCE of order $P = 1$ with $M = 3$ random variables and the sparse Gauss-Hermite grid with $n_q = 25$ points.

	mean	st. dev. σ	σ/mean
<i>CL</i>	0.8552	0.0049	0.0058
<i>CD</i>	0.0183	0.00012	0.0065

where $\kappa_1 = \kappa((x_1, 0, z_1), \omega)$, $\kappa_2 = \kappa((x_2, 0, z_2), \omega)$ are two random variables in points $(x_1, 0, z_1)$ and $(x_2, 0, z_2)$. For numerical simulations we take the covariance lengths $l_1 = |\max_i(x_i) - \min_i(x_i)|/10$ and $l_2 = |\max_i(z_i) - \min_i(z_i)|/10$, standard deviation $\sigma = 10^{-3}$, $m = 3$ the number of KLE terms (see Eq. 6), the stochastic dimension $M = 3$ and the number of sparse Gauss-Hermite points (in 3D) for computing PCE coefficients in (Eq. 8) $n_q = 25$. In [13] one can see 21 random realisations of RAE-2822 airfoil.

Table 1 demonstrates the surprisingly small uncertainties (the last column) in the lift and in the drag — 0.58% and 0.65% correspondingly. A possible explanation can be small uncertain perturbations in the airfoil geometry.

3 Discretisation Techniques

In the following, $(\Omega, \mathcal{B}, \mathbb{P})$ denotes a probability space, where Ω is the set of elementary events, \mathcal{B} is the σ -algebra of events and \mathbb{P} is the probability measure. The symbol ω always specifies an elementary event $\omega \in \Omega$.

The random field $\kappa(x, \omega)$ needs to be discretised both in the stochastic and in the spatial dimensions. One of the main tools here is the Karhunen-Loève expansion (KLE) [16]. By definition, KLE of a random field $\kappa(x, \omega)$ is the following series [16]

$$\kappa(x, \omega) = \bar{\kappa}(x) + \sum_{\ell=1}^{\infty} \sqrt{\lambda_{\ell}} \phi_{\ell}(x) \xi_{\ell}(\omega), \tag{6}$$

where $\xi_{\ell}(\omega)$ are uncorrelated random variables and $\bar{\kappa}(x)$ is the mean value of $\kappa(x, \omega)$, λ_{ℓ} and ϕ_{ℓ} are the eigenvalues and the eigenvectors of problem

$$T \phi_{\ell} = \lambda_{\ell} \phi_{\ell}, \quad \phi_{\ell} \in L^2(\mathcal{G}), \ell \in \mathbb{N}, \tag{7}$$

and operator T is defined like follows

$$T : L^2(\mathcal{G}) \rightarrow L^2(\mathcal{G}), \quad (T\phi)(x) := \int_{\mathcal{G}} \text{cov}_{\kappa}(x, y) \phi(y) dy,$$

where cov_{κ} is a given covariance function. Throwing away all unimportant terms in KLE, one obtains the truncated KLE, which is a sparse representation of the random field $\kappa(x, \omega)$. Each random variable ξ_{ℓ} can be approximated in a set of

new independent Gaussian random variables (polynomial chaos expansions (PCE) of Wiener [7, 25]), e.g.

$$\xi_\ell(\omega) = \sum_{\beta \in \mathcal{J}} \xi_\ell^{(\beta)} H_\beta(\theta(\omega)),$$

where $\theta(\omega) = (\theta_1(\omega), \theta_2(\omega), \dots), \xi_\ell^{(\beta)}$ are coefficients, H_β are multivariate Hermite polynomials, $\beta \in \mathcal{J}$ is a multiindex, $\mathcal{J} := \{\beta \mid \beta = (\beta_1, \dots, \beta_j, \dots), \beta_j \in \mathbb{N}_0\}$ is a multi-index set [18].

For the purpose of actual computation, truncate the polynomial chaos expansion after finitely many terms, e.g.

$$\beta \in \mathcal{J}_{M,P} := \{\beta \in \mathcal{J} \mid \gamma(\beta) \leq M, \quad |\beta| \leq P\}, \quad \gamma(\beta) := \max\{j \in \mathbb{N} \mid \beta_j > 0\}.$$

Since Hermite polynomials are orthogonal, the coefficients $\xi_\ell^{(\beta)}$ can be computed by projection

$$\xi_\ell^{(\beta)} = \frac{1}{\beta!} \int_{\Theta} H_\beta(\theta) \xi_\ell(\theta) \mathbb{P}(d\theta).$$

This multidimensional integral over Θ can be computed approximately, for example, on a sparse Gauss-Hermite grid with n_q grid points

$$\xi_\ell^{(\beta)} \approx \frac{1}{\beta!} \sum_{i=1}^{n_q} H_\beta(\theta_i) \xi_\ell(\theta_i) w_i, \tag{8}$$

where weights w_i and points θ_i are defined from sparse Gauss-Hermite integration rule. After a finite element discretisation (see [10] for more details) the discrete eigenvalue problem (7) looks like

$$\mathbf{MCM}\phi_\ell = \lambda_\ell^h \mathbf{M}\phi_\ell, \quad \mathbf{C}_{ij} = \text{cov}_\kappa(x_i, y_j). \tag{9}$$

Here the mass matrix \mathbf{M} is stored in a usual data sparse format and the dense matrix $\mathbf{C} \in \mathbb{R}^{n \times n}$ (requires $\mathcal{O}(n^2)$ units of memory) is approximated in the sparse \mathcal{H} -matrix format [10] (requires only $\mathcal{O}(n \log n)$ units of memory) or in the Kronecker low-rank tensor format [9]. To compute m eigenvalues ($m \ll n$) and corresponding eigenvectors we apply the Lanczos eigenvalue solver [11, 22].

4 Low-Rank Response Surface

To compute statistics of the random (uncertain) solution (error-bars, quantiles, cumulative density function, etc) accurate enough, one needs a large sample size. Monte Carlo simulations are expensive. To decrease the computational costs we compute a, so-called, response surface — (multivariate) polynomial (see Eq. (10)). The idea [13] is to construct a good response surface from few samples and then

to use the residual for its improvement. A motivation for this idea comes from the fact that in many software packages for solving engineering and physical problems it is impossible or very difficult to change the code, but it is possible to access the residual. Later on the computed response surface is used for very fast generation of a large sample.

Let $\mathbf{v}(x, \theta)$ be the solution (or a functional of the solution). It can be pressure, density, velocity, lift, drag etc. $\mathbf{v}(x, \theta)$ can be approximated in a set of new independent Gaussian random variables (truncated polynomial chaos expansions of Wiener [25])

$$\mathbf{v}(x, \theta(\omega)) \approx \sum_{\beta \in \mathcal{J}_{M,P}} \mathbf{v}_\beta(x) H_\beta(\theta) = [\dots \mathbf{v}_\beta(x) \dots] [\dots H_\beta(\theta) \dots]^T, \quad (10)$$

where coefficients $\mathbf{v}_\beta(x)$ are computed as follows

$$\mathbf{v}_\beta(x) = \frac{1}{\beta!} \int_{\Theta} H_\beta(\theta) \mathbf{v}(x, \theta) \mathbb{P}(d\theta) \approx \frac{1}{\beta!} \sum_{i=1}^{n_q} H_\beta(\theta_i) \mathbf{v}(x, \theta_i) w_i, \quad (11)$$

The PCE representation in Eq. 10 was used to compute the mean and the variance of the pressure (see Fig. 3) for the Case 1 ($\alpha = 1.93$ and $\text{Ma} = 0.676$, no shock). PCE coefficients are computed by the sparse Gauss Hermite grid with $n_q = 281$ nodes. Here the multidimensional integral over Θ is computed approximately, for example, on a sparse Gauss-Hermite grid [6, 2].

Fig. 4 demonstrates the mean of density and mean of pressure, computed again as in Eq. 10 for the Case 9 ($\alpha = 2.79$ and $\text{Ma} = 0.734$, with shock).

Fig. 5 demonstrates the variance of density and variance of pressure, computed via Monte Carlo methods for the Case 9. One can see the largest uncertainty in the shock position.

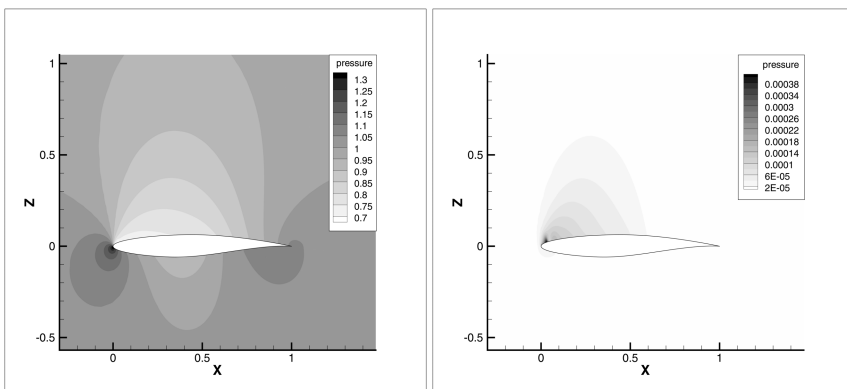


Fig. 3 (left) The mean pressure in Case 1; (right) The variance of the pressure in Case 1. Both are computed by the sparse Gauss Hermite grid with $n_q = 281$ nodes.

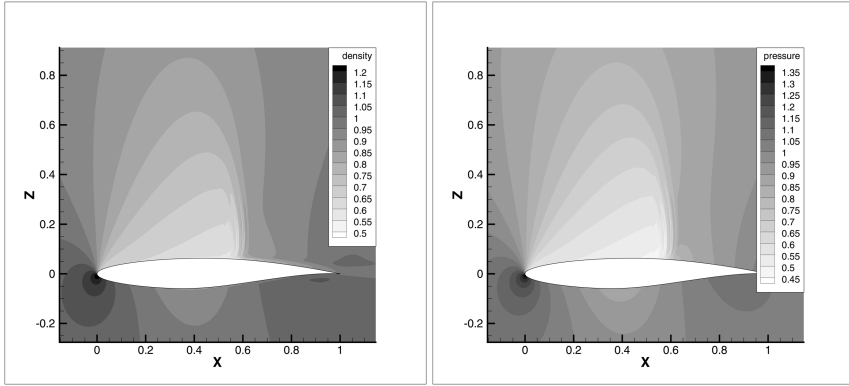


Fig. 4 (left) The mean density; (right) the mean pressure computed by PCE in the Case 9

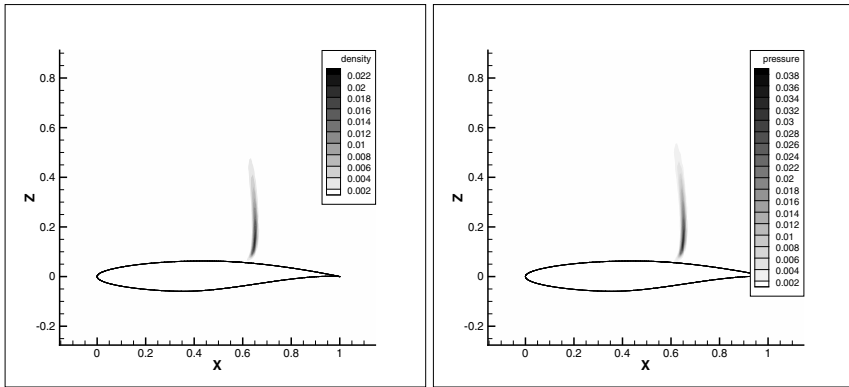


Fig. 5 (left) The variance of the density; (right) The variance of the pressure computed by MC in the Case 9.

Using the rank- k approximation of $[\mathbf{v}(x, \theta_1), \dots, \mathbf{v}(x, \theta_{n_q})]$, obtain

$$\mathbf{v}_\beta(x) = \frac{1}{\beta!} [\mathbf{v}(x, \theta_1), \dots, \mathbf{v}(x, \theta_{n_q})] \cdot [H_\beta(\theta_1)w_1, \dots, H_\beta(\theta_{n_q})w_{n_q}]^T \approx \mathbf{A}\mathbf{B}^T \mathbf{c}_\beta, \quad (12)$$

where $\mathbf{A} \in \mathbb{R}^{n \times k}$, $\mathbf{B} \in \mathbb{R}^{n_q \times k}$, $k \ll \min\{n, n_q\}$ and vector $\mathbf{c}_\beta := \frac{1}{\beta!} [H_\beta(\theta_1)w_1, \dots, H_\beta(\theta_{n_q})w_{n_q}]^T$. The matrix of all PCE coefficients will be

$$[\dots \mathbf{v}_\beta(x) \dots] = \mathbf{A}\mathbf{B}^T [\dots \mathbf{c}_\beta \dots], \quad \beta \in \mathcal{I}_{M,P}. \quad (13)$$

Taking Eq. [10](#) and Eq. [13](#), obtain the final formula for the **low-rank response surface**

$$\mathbf{v}(x, \theta(\omega)) \approx \mathbf{A}\mathbf{B}^T [\dots \mathbf{c}_\beta \dots] [\dots H_\beta(\theta) \dots]^T. \quad (14)$$

4.1 Update of the Low-Rank Response Surface via Computing the Residual

In real-world applications, the deterministic solver is very complicated and it is difficult or even impossible to change it, but one can often print out the norm of the residual. Assume that we already approximated the unknown solution by a response surface. Our response surface is approximation via multivariate Hermite polynomials like in Eq. [14](#) where coefficients are computed like in Eq. [13](#) with quadrature points $\theta_i, i = 1..n_q$. The following algorithm updates the given response surface.

Algorithm: (Update of the response surface)

1. Take the next point θ_{n_q+1} and evaluate the response surface Eq. [14](#) in this point. Let $u(x, \theta_{n_q+1})$ be the obtained predicted solution.
2. Compute the norm of the residual $\|\mathbf{r}\|$ of the deterministic problem (e.g. evaluate one iteration). If $\|\mathbf{r}\|$ is small then there is no need to solve the expensive deterministic problem in θ_{n_q+1} , otherwise (if $\|\mathbf{r}\|$ is large) solve the deterministic problem and recompute \mathbf{A}, \mathbf{B}^T and \mathbf{c}_β in Eq. [12](#).
3. Go to item (1).

In the best case we never solve the deterministic problem again. In the worst case we must solve the deterministic problem for each $\theta_{n_q+i}, i = 1, 2, \dots$. To test this algorithm we computed the solution in Case 1 with 10000 TAU iterations (is usual number of iterations). Then, first, we computed the solution with the response surface (as described above) and, second, corrected it with 1000 TAU iterations. Then we compared both solutions and observed only a very small difference. Thus, the response surface reduced the number of needed iterations from 10000 to 1000. We note that the solution in Case 1 is smooth and there is no shock.

We tested this Algorithm also in the Case 9 (solution with a shock) and it failed. We pre-computed the solution by two different response surfaces (of order $P = 2$ and $P = 4$). Both response surfaces failed to produce a good result. For instance, we observed not only one shock, but many smaller shocks. Then we observed an increasing range of e.g. pressure (range $(-6; 5)$ in contrast to $(0.5, 1.3)$). It is similar when one tries to approximate a step function by a polynomial — the amplitude of oscillations grows up. Another negative effect which we observed during further iterating the solution, obtained from the response surface, was that the deterministic solver (TAU) produces “nan” after few iterations. A possible reason is that some important solution values, obtained from the response surface, are out of the physical range (e.g. negative density) and are non-realistic.

Thus, we can come to the conclusion that if the solution is smooth (e.g. as in Case 1) then response surface produces a good starting value. Otherwise, if the solution has a shock, the response surface produces a very poor approximation and further iterations do not help.

The computed solution $u(x, \theta_{n_q+1})$ can be used to update the response surface, i.e. to recompute the matrices \mathbf{A}, \mathbf{B} and $[\dots\mathbf{c}_\beta\dots]$ and PCE coefficients (Eq. [13](#)). Please note that this update works only in the case of the usage of embedded sparse grids or (Q)MC in Eq. [11](#).

5 Data Compression

A large number of stochastic realisations of random fields requires a large amount of memory and powerful computational resources. To decrease memory requirements and computing time we offer to use a low-rank approximation for all realisations of the solution [13]. This low-rank approximation allows us an effective postprocessing (computation of the mean value, variance, exceedance probability) with drastically reduced memory requirements (see Table 4). For each new realisation only the corresponding low-rank update will be computed (see, e.g. [11]). This can be practical when, e.g. the results of many thousands Monte Carlo simulations should be computed and stored. Let $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1..Z$, be the solution vector (already centred), where Z is a number of stochastic realisations of the solution. Let us build from all these vectors the matrix $\mathbf{W} = (\mathbf{v}_1, \dots, \mathbf{v}_Z) \in \mathbb{R}^{n \times Z}$ and consider the factorisation

$$\mathbf{W} = \mathbf{A}\mathbf{B}^T, \quad \text{where } \mathbf{A} \in \mathbb{R}^{n \times k} \quad \text{and} \quad \mathbf{B} \in \mathbb{R}^{Z \times k}. \quad (15)$$

We say that matrix \mathbf{W} is a **rank- k matrix** if the representation in Eq. 15 is given. We denote the class of all rank- k matrices for which factors \mathbf{A} and \mathbf{B}^T in Eq. 15 exist by $\mathcal{R}(k, n, Z)$. If $\mathbf{W} \in \mathcal{R}(k, n, Z)$ we say that \mathbf{W} has a **low-rank representation**. The first aim is to compute a rank- k approximation \mathbf{W}_k of \mathbf{W} , such that

$$\|\mathbf{W} - \mathbf{W}_k\| < \varepsilon, \quad k \ll \min\{n, Z\}.$$

The second aim is to compute an update for the approximation \mathbf{W}_k with a linear complexity for every new coming vector \mathbf{v}_{Z+1} . Below in Section 5.1 we present the algorithm which performs this.

To get the reduced singular value decomposition we omit all singular values, which are smaller than a given level ε or, alternative variant, we leave a fixed number of largest singular values. After truncation we speak about *reduced singular value decomposition* (denoted by *rSVD*) $\mathbf{W}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T$, where $\mathbf{U}_k \in \mathbb{R}^{n \times k}$ contains the first k columns of \mathbf{U} , $\mathbf{V}_k \in \mathbb{R}^{Z \times k}$ contains the first k columns of \mathbf{V} and $\Sigma_k \in \mathbb{R}^{k \times k}$ contains the k -biggest singular values of Σ .

The computation of such basic statistics as the mean value, the variance, the exceedance probability can be done with a linear complexity. The following examples illustrate computation of the mean value and the variance.

Let us take $\mathbf{A} := \mathbf{U}_k \Sigma_k$ and $\mathbf{B}^T := \mathbf{V}_k^T \in \mathbb{R}^{k \times Z}$. Denote the j -th row of matrix \mathbf{A} by $\mathbf{a}_j \in \mathbb{R}^k$ and the i -th column of matrix \mathbf{B}^T by $\mathbf{b}_i \in \mathbb{R}^k$. It is evident, that if \mathbf{W} is given explicitly, one can compute the mean value and the variance just keeping in memory 2 vectors - the mean (variance) and the current value. Below we show how to compute the mean and the variance if only \mathbf{A} and \mathbf{B} are given.

1. One can compute the mean solution $\bar{\mathbf{v}} \in \mathbb{R}^n$ as follows

$$\bar{\mathbf{v}} = \frac{1}{Z} \sum_{i=1}^Z \mathbf{v}_i = \frac{1}{Z} \sum_{i=1}^Z \mathbf{A} \cdot \mathbf{b}_i = \mathbf{A} \bar{\mathbf{b}}, \quad (16)$$

Table 2 Rank- k approximation errors of the mean and of the variance of density in Case 1.

rank k	5	20
$\max_x \bar{\rho}(x) - \bar{\rho}_k(x) $	1.7e-6	4.2e-10
$\max_x \text{var}(\rho)(x) - \text{var}(\rho)_k(x) $	6.7e-5	2.3e-8

The computational complexity is $\mathcal{O}(k(Z+n))$, in contrast to $\mathcal{O}(nZ)$ for usual dense data format. As a demonstration we compute the mean.

2. One can compute the variance of the solution $\text{var}(\mathbf{v}) \in \mathbb{R}^n$ by the computing the covariance matrix and taking its diagonal. First, one computes the centred matrix

$$\mathbf{W}_c := \mathbf{W} - \bar{\mathbf{v}} \mathbf{1}^T, \quad \text{where } \bar{\mathbf{v}} = \mathbf{W} \cdot \mathbf{1} / Z, \text{ and } \mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^Z. \quad (17)$$

Computing \mathbf{W}_c costs $\mathcal{O}(k^2(n+Z))$ (addition and truncation of rank- k matrices). By definition, the covariance matrix is $\mathbf{C} = \frac{1}{Z-1} \mathbf{W}_c \mathbf{W}_c^T$. The reduced singular value decomposition of \mathbf{W}_c is $(\mathbf{W}_c)_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^T$, $\mathbf{U}_k \in \mathbb{R}^{n \times k}$, $\Sigma_k \in \mathbb{R}^{k \times k}$ and $\mathbf{V}_k \in \mathbb{R}^{Z \times k}$ can be computed via the QR algorithm [5, 13]. Now, the covariance matrix can be written like

$$\mathbf{C} = \frac{1}{Z-1} (\mathbf{W}_c)_k (\mathbf{W}_c)_k^T \approx \frac{1}{Z-1} \mathbf{U}_k \Sigma_k \Sigma_k^T \mathbf{U}_k^T. \quad (18)$$

The variance of the solution vector (i.e. the diagonal of the covariance matrix \mathbf{C}) can be computed with the complexity $\mathcal{O}(k^2(Z+n))$.

Table 2 demonstrates the rank-5 and rank-20 approximations of the mean and of the variance of density. One can see that both rank- k approximation errors are very small, much smaller than e.g. the discretisation error or Monte Carlo error (by computing the mean value).

Lemma 0.1. *Let $\|\mathbf{W} - \mathbf{W}_k\|_2 \leq \varepsilon$, and $\bar{\mathbf{u}}_k$ be a rank- k approximation of the mean $\bar{\mathbf{u}}$. Then a) $\|\bar{\mathbf{u}} - \bar{\mathbf{u}}_k\| \leq \frac{\varepsilon}{\sqrt{N_s}}$, b) $\|\mathbf{W}_c - (\mathbf{W}_c)_k\| \leq \varepsilon$, c) $\|\mathbf{C} - \mathbf{C}_k\| \leq \frac{1}{N_s-1} \varepsilon^2$.*

Proof: Since $\bar{\mathbf{u}} = \frac{1}{N_s} \mathbf{W} \mathbf{1}$ and $\bar{\mathbf{u}}_k = \frac{1}{N_s} \mathbf{W}_k \mathbf{1}$, then

$$\|\bar{\mathbf{u}} - \bar{\mathbf{u}}_k\|_2 = \frac{1}{N_s} \|(\mathbf{W} - \mathbf{W}_k) \mathbf{1}\|_2 \leq \frac{1}{N_s} \|(\mathbf{W} - \mathbf{W}_k)\|_2 \cdot \|\mathbf{1}\|_2 \leq \frac{\varepsilon}{\sqrt{N_s}}.$$

Let $\mathbf{I} \in \mathbb{R}^{N_s \times N_s}$ be the identity matrix, then

$$\begin{aligned} \|\mathbf{W}_c - (\mathbf{W}_c)_k\|_2 &\leq \|\mathbf{W} - \mathbf{W}_k\|_2 \cdot \left\| \mathbf{I} - \frac{1}{N_s} \cdot \mathbf{1} \cdot \mathbf{1}^T \right\|_2 \leq \varepsilon, \quad \text{and} \\ \|\mathbf{C} - \mathbf{C}_k\|_2 &\leq \frac{1}{N_s-1} \|\mathbf{W}_c \mathbf{W}_c^T - (\mathbf{W}_c)_k (\mathbf{W}_c)_k^T\|_2 \\ &= \frac{1}{N_s-1} \|\mathbf{U} \Sigma \Sigma^T \mathbf{U}^T - \mathbf{U}_k \Sigma_k \Sigma_k^T \mathbf{U}_k^T\| \leq \frac{1}{N_s-1} \varepsilon^2. \end{aligned}$$

5.1 Concatenation of Two Low-Rank Matrices

Let \mathbf{A} and \mathbf{B} such that $\mathbf{W}_k = \mathbf{A}\mathbf{B}^T$ be given. Suppose also that matrix $\mathbf{W}' \in \mathbb{R}^{n \times m}$ contains new m solution vectors. For a small m , computing the factors $\mathbf{C} \in \mathbb{R}^{n \times k}$ and $\mathbf{D} \in \mathbb{R}^{m \times k}$ such that $\mathbf{W}' \approx \mathbf{C}\mathbf{D}^T$ is not expensive. Now our purpose is to compute with a linear complexity the rank- k approximation of $\mathbf{W}_{\text{new}} := [\mathbf{W} \ \mathbf{W}'] \in \mathbb{R}^{n \times (Z+m)}$. To do this, we build two concatenated matrices $\mathbf{A}_{\text{new}} := [\mathbf{A} \ \mathbf{C}] \in \mathbb{R}^{n \times 2k}$ and $\mathbf{B}_{\text{new}}^T = \text{blockdiag}[\mathbf{B}^T \ \mathbf{D}^T] \in \mathbb{R}^{2k \times (Z+m)}$. Note that the difficulty now is that matrices \mathbf{A}_{new} and \mathbf{B}_{new} have rank $2k$. The rank k approximation of the new matrix \mathbf{W}_{new} is done with a linear complexity $\mathcal{O}((n+Z)k^2 + k^3)$ (for details see [13]).

6 Bayesian Update of the Uncertain Airfoil Geometry

We assume that the airfoil geometry contains random deformations (e.g. dents). A possible reason, for example, can be the influence of external forces. First our task is to parametrize all such deformations for all given airfoils. We offer to use random fields $\kappa(x, \omega)$, where ω is a vector of random parameters (see Section 2.2). The problem is that the probability density function of ω is unknown. We assume it a priori as Gaussian. If we could measure all given airfoils (from different airplanes) then we could build a good parametrization model, but everything we can do is to measure airfoils only in a few points. This is our knowledge. The question now is how to incorporate this knowledge to our parametrization model? We can do this by using the Bayesian update. The Algorithm is described in [21, 20].

In Fig. 6 (left) you may see:

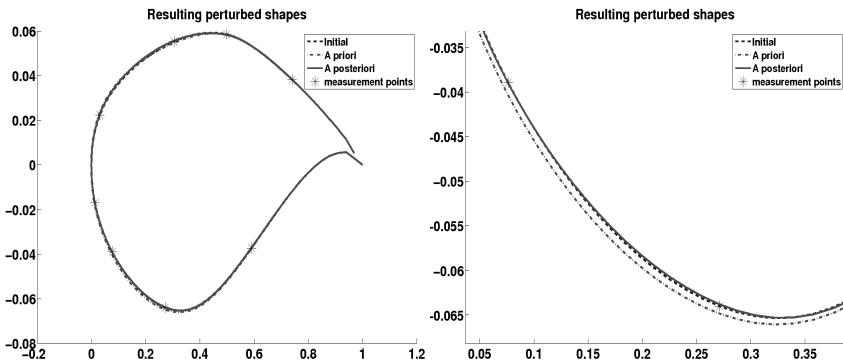


Fig. 6 (left) The truth airfoil (is in reality unknown), a priori (is our initial assumption) and a posteriori (the measurements are taken into account) airfoils. (right) Detailed RAE-2822 airfoil picture in interval [0.05, 0.35].

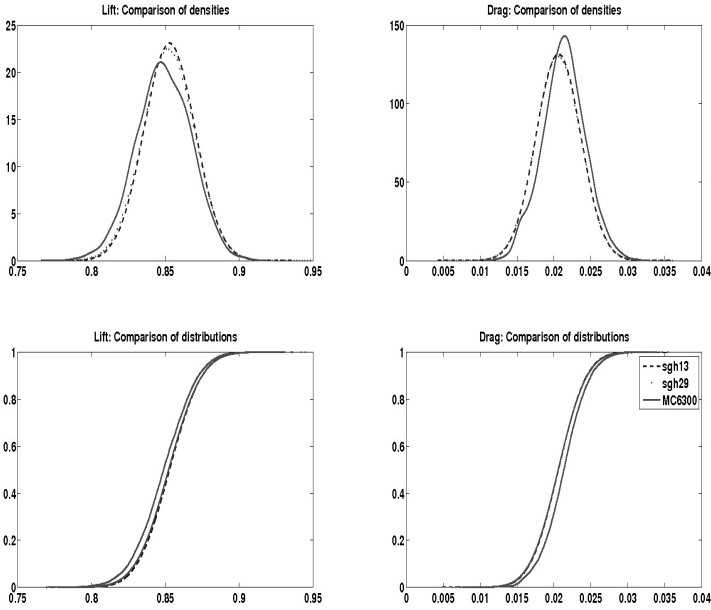


Fig. 7 Probability density functions (first row), cumulative distribution functions (second row) of *CL* (left) and *CD* (right). PCE is of order 1 with two random variables. Three graphics computed with 6360 MC simulations, $n_q = 13$ and $n_q = 29$ collocation points.

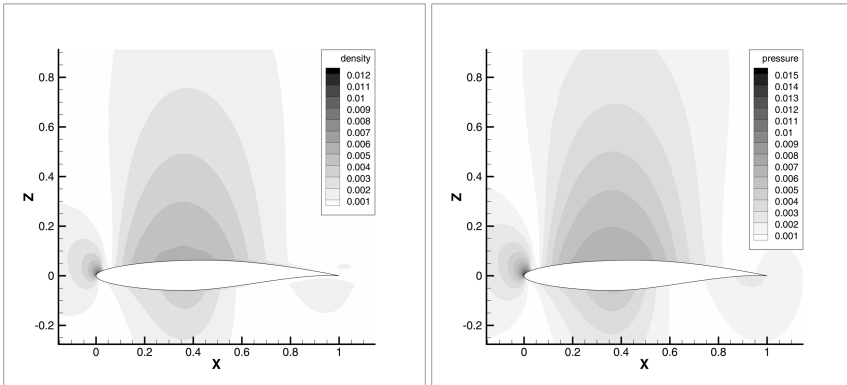


Fig. 8 (left) $\|\overline{\rho_{MC}} - \overline{\rho_{SGH}}\|$ and (right) $\|\overline{p_{MC}} - \overline{p_{SGH}}\|$. p_{SGH} was computed from the sparse Gauss Hermite grid with 281 nodes. $\rho_{SGH} \in (0.65, 1.2)$, $p_{SGH} \in (0.7, 1.3)$. Case 1.

sparse Gauss Hermite grid. One can see that the difference is very small compared to the corresponding physical values ρ_{SGH} and p_{SGH} .

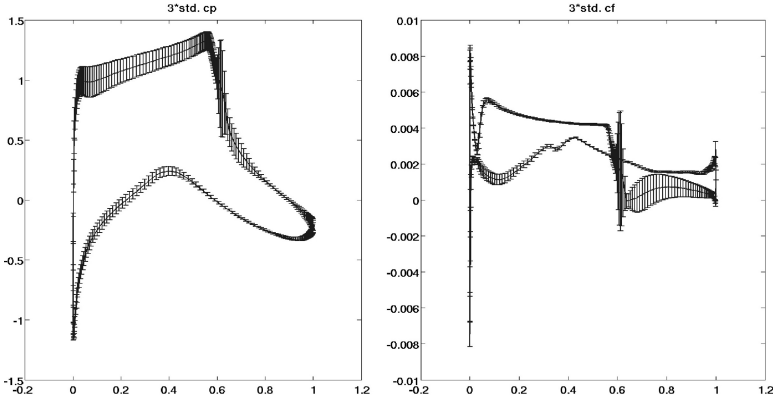


Fig. 9 3σ error bars in each point of the RAE-2822 airfoil surface for the pressure coefficient cp (left) and friction coefficient cf (right).

The graphics in Fig. 9 demonstrate 3σ error bars, σ the standard deviation, for the pressure cp and absolute skin friction cf coefficients in each surface point of the RAE-2822 airfoil. The data are obtained from 645 realisation of the solution. One can see that the largest uncertainty occurs at the shock ($x \approx 0.6$). A possible explanation is that the shock position is expected to change slightly with varying parameters α and Ma .

In Table 4 one can see relative errors of the rank- k approximations (in the Frobenius norm). Five solution matrices contain pressure, density, turbulence kinetic energy (tke), turbulence omega (to) and eddy viscosity (ev) in the whole computational domain with 260000 dofs. Additionally, one can also see much smaller memory requirement (dense matrix format costs 1.25GB). The column 7 shows the computing time required for the SVD-update (the Algorithm described in Section 5.1) and the the column 8 time required for the full SVD of the global matrix $\in \mathbb{R}^{260000 \times 600}$ correspondingly. A possible explanation for the large computing time for the full SVD is the lack of memory and expensive swapping of data.

Table 4 Relative errors and computational requirements of rank- k approximations of the solution matrices $\in \mathbb{R}^{260000 \times 600}$. Memory required for the storage of each matrix in the dense matrix format is 1.25 GB.

rank k	pressure	density	tke	to	ev	time, sec update	time, sec full SVD	memory MB
10	1.9e-2	1.9e-2	4.0e-3	1.4e-3	1.4e-3	107	1537	21
20	1.4e-2	1.3e-2	5.9e-3	3.3e-4	4.1e-4	150	2084	42
50	5.3e-3	5.1e-3	1.5e-4	9.1e-5	7.7e-5	228	8236	104

Table 5 Rank- k approximation errors of the variance of pressure and of the variance of density in Case 9

rank k	5	30
$\max_x \text{var}(p)(x) - \text{var}(p)_k(x) $	5.3e-3	1.6e-4
$\max_x \text{var}(\rho)(x) - \text{var}(\rho)_k(x) $	3.5e-3	8.8e-5

In Table 5 we provide the rank $k = \{5, 30\}$ approximation errors (in the maximum norm) of the variance of the pressure and of the density (compare with Fig. 5). The variances $\text{var}(p)_k(x)$ and $\text{var}(\rho)_k(x)$ were computed from the matrix $\mathbf{W} \in \mathbb{R}^{65568 \times 1521}$ as described in Section 5.

Further, we consider Case 1 ($\alpha = 1.93$ and $\text{Ma} = 0.676$, no shock). Fig. 10 shows relative errors (for the Case 1) in the Frobenius and the maximum norms for pressure and density computed in 10 points of a two-dimensional sparse Gauss-Hermite grid. These relative errors compare the solution which we obtain after 10000 TAU iterations without any start value with the solution which we obtain after only 2000 TAU iterations with start values taken from the response surface (multivariate Hermite polynomials with $M = 2$ variables and of order $P = 2$). One can see that the errors are very small (of order 10^{-3}), i. e. the response surface produces a good approximation. We note that 10 chosen points are lying in a small neighbourhood of the point $\alpha = 1.93$ and $\text{Ma} = 0.676$.

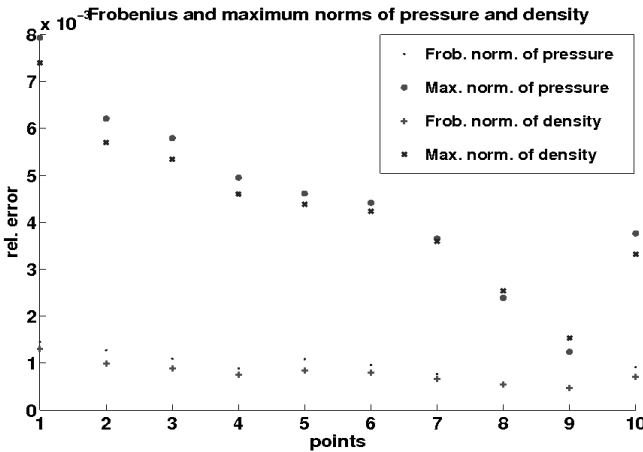


Fig. 10 Relative errors (Case 1) in the Frobenius and the maximum norms for the pressure and density

References

- [1] Brand, M.: Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra Appl.* 415(1), 20–30 (2006)
- [2] Bungartz, H.-J., Griebel, M.: Sparse grids. *Acta Numer.* 13, 147–269 (2004)
- [3] Chen, Q.-Y., Gottlieb, D., Hesthaven, J.S.: Uncertainty analysis for the steady-state flows in a dual throat nozzle. *Journal of Computational Physics* 204(1), 378–398 (2005)
- [4] Gerhold, T., Friedrich, O., Evans, J., Galle, M.: Calculation of complex three-dimensional configurations employing the DLR-tau-Code. AIAA-0167 (1997)
- [5] Golub, G.H., Van Loan, C.F.: *Matrix computations*, 3rd edn. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore (1996)
- [6] Griebel, M.: Sparse grids and related approximation schemes for higher dimensional problems. In: *Foundations of Computational Mathematics, Santander 2005*. London Math. Soc. Lecture Note Ser., vol. 331, pp. 106–161. Cambridge Univ. Press, Cambridge (2006)
- [7] Hida, T., Kuo, H.-H., Potthoff, J., Streit, L.: *White noise - An infinite-dimensional calculus. Mathematics and its Applications*, vol. 253. Kluwer Academic Publishers Group, Dordrecht (1993)
- [8] Hosder, S., Walters, R., Perez, R.: A non-intrusive polynomial chaos method for uncertainty propagation in cfd simulations. In: *44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, (AIAA 2006-891)*, pp. 209–236 (January 2006)
- [9] Khoromskij, B.N., Litvinenko, A.: Data sparse computation of the Karhunen-Loève expansion. In: *Numerical Analysis and Applied Mathematics: International Conference on Numerical Analysis and Applied Mathematics, AIP Conf. Proc.*, vol. 1048(1), pp. 311–314 (2008)
- [10] Khoromskij, B.N., Litvinenko, A., Matthies, H.G.: Application of hierarchical matrices for computing the Karhunen-Loève expansion. *Computing* 84(1-2), 49–67 (2009)
- [11] Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Research Nat. Bur. Standards* 45, 255–282 (1950)
- [12] Litvinenko, A., Matthies, H.G.: Sparse data representation of random fields. In: *Proceedings in Applied Mathematics and Mechanics, PAMM*, vol. 9, pp. 587–588. Wiley-InterScience (2009)
- [13] Litvinenko, A., Matthies, H.G.: Low-rank data format for uncertainty quantification. In: Skiadas, C.H. (ed.) *International Conference on Stochastic Modeling Techniques and Data Analysis Proceedings*, Chania, Greece, pp. 477–484 (2010), <http://www.smta.net/smta2010proceedings.html>
- [14] Litvinenko, A., Matthies, H.G.: Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics. In: *IV European Congress on Computational Mechanics (ECCM IV): Solids, Structures and Coupled Problems in Engineering* (2010), http://www.eccm2010.org/complet/fullpaper_1036.pdf
- [15] Litvinenko, A., Matthies, H.G.: Uncertainties quantification and data compression in numerical aerodynamics. *Proc. Appl. Math. Mech.* 11(1), 877–878 (2011)
- [16] Loève, M.: *Probability theory I. Graduate Texts in Mathematics*, 4th edn., vol. 45, 46. Springer, New York (1977)
- [17] Mathelin, L., Hussaini, M.Y., Zang, T.A.: Stochastic approaches to uncertainty quantification in CFD simulations. *Numer. Algorithms* 38(1-3), 209–236 (2005)

- [18] Matthies, H.G.: Uncertainty quantification with stochastic finite elements. Part 1. Fundamentals. In: *Encyclopedia of Computational Mechanics*. John Wiley and Sons, Ltd. (2007)
- [19] Najm, H.N.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. In: *Annual Review of Fluid Mechanics*. *Annu. Rev. Fluid Mech.*, vol. 41, pp. 35–52. Annual Reviews, Palo Alto (2009)
- [20] Pajonk, O., Rosić, B.V., Litvinenko, A., Matthies, H.G.: A deterministic filter for non-gaussian bayesian estimation-applications to dynamical system estimation with noisy measurements. *Physica D: Nonlinear Phenomena* 241(7), 775–788 (2012)
- [21] Rosic, B.V., Litvinenko, A., Pajonk, O., Matthies, H.G.: Sampling-free linear bayesian update of polynomial chaos representations. *J. Comput. Physics* 231(17), 5761–5787 (2012)
- [22] Saad, Y.: *Numerical methods for large eigenvalue problems*. Algorithms and Architectures for Advanced Scientific Computing. Manchester University Press, Manchester (1992)
- [23] Simon, F., Guillen, P., Sagaut, P., Lucor, D.: A gpc-based approach to uncertain transonic aerodynamics. *Computer Methods in Applied Mechanics and Engineering* 199(17-20), 1091–1099 (2010)
- [24] Wan, X., Karniadakis, G.E.: Long-term behavior of polynomial chaos in stochastic flow simulations. *Computer Methods in Applied Mechanics and Engineering* 195(41-43), 5582–5596 (2006); John H. Argyris Memorial Issue. Part II
- [25] Wiener, N.: The homogeneous chaos. *American Journal of Mathematics* 60, 897–936 (1938)
- [26] Witteveen, J.A.S., Loeven, A., Bijl, H.: An adaptive stochastic finite elements approach based on newton-cotes quadrature in simplex elements. *Computers & Fluids* 38(6), 1270–1288 (2009)
- [27] Witteveen, J.A.S., Doostan, A., Pecnik, R., Iaccarino, G.: Uncertainty quantification of the transonic flow around the rae2822 airfoil. *Annual Research Briefs, Center for Turbulence Research*, pp. 93–104. Stanford University (2009)

Efficient Quantification of Aerodynamic Uncertainties Using Gradient-Employing Surrogate Methods

Dishi Liu

Abstract. Uncertainty quantification (UQ) in aerodynamic simulations is hindered by the high computational cost of CFD models. With gradient information obtained efficiently by using an adjoint solver, gradient-employing surrogate methods are promising in speeding up the UQ process. To investigate the efficiency of UQ methods we apply gradient-enhanced radial basis functions, gradient-enhanced point-collocation polynomial chaos, gradient-enhanced Kriging and quasi-Monte Carlo (QMC) quadrature to a test case where the geometry of an RAE2822 airfoil is perturbed by a Gaussian random field parameterized by 10 independent variables. The four methods are compared in their efficiency in estimating some statistics and the probability distribution of the uncertain lift and drag coefficients. The results show that with the same computational effort the gradient-employing surrogate methods achieve better accuracy than the QMC does.

1 Introduction

In aerodynamic simulations it is beneficial to consider uncertainties in the inputs, the formulation and the numerical error of the CFD model. In this work our concern is confined to the uncertainties in the model's input and probabilistic approaches for uncertainty quantification (UQ) for CFD models. The uncertainties in the input propagates to the system response quantities (SRQ) through the model. Minor uncertainties can have an amplified impact in some instances and lead to occurrences of rare catastrophic events. Quantifying the uncertainties associated with the SRQ enhances the reliability of the simulations and enables robust design optimization. Most often this UQ process is done in a probabilistic framework in which the input

Dishi Liu

German Aerospace Center (DLR), Institute für Aerodynamik und Strömungstechnik,
Lilienthalplatz 7, 38108 Braunschweig, Germany

e-mail: dishi.liu@dlr.de

uncertainties are represented by random variables, and the consequent uncertainties in the SRQ are quantified by determining its probability distribution or statistical moments.

However, uncertainties in the input, especially those spatially or temporally distributed, like geometric uncertainties, often generate a large number of variables. The “curse of dimensionality” prohibits the use of tensor-product quadratures. In [16] and [21] sparse grid quadratures were employed in aerodynamic UQ problems due to uncertain airfoil geometry. Nevertheless, if the number of variables is larger than 10 even sparse grid methods suffer limitations in applicability [17]. The high computational cost of CFD models also makes the traditional sampling methods such as Monte-Carlo and its variance-reduced variants (e.g. Latin Hypercube method) not efficient due to their slower error convergence rate.

Surrogate methods are gaining more attention in UQ as they provide approximations of the CFD model which are much cheaper to evaluate while maintaining a reasonable accuracy so that the UQ can be performed on the basis of a large number of samples evaluated on the surrogate model. E.g. [12] shows a Kriging surrogate method better than plain Monte Carlo and Latin Hypercube methods in estimating the mean value of a bivariate Rosenbrock function. A comparative study of surrogate methods that are not employing gradients [23] shows Kriging is more accurate than radial basis functions and multivariate polynomial in approximating some 10-variate test functions.

Gradient-employing also give an edge to surrogate methods if the gradients are obtained at a relatively lower cost than that of the SRQ, which is the case when an adjoint CFD solver [5] is used and the number of SRQ is less than the number of variables. It should be noted that the gradient information cannot be effectively utilized by the UQ methods based on direct sampling of the CFD model. A naive augmentation of samples by finite difference brings no benefit because the augmenting samples are not statistically independent of the original ones.

Different sampling schemes are adopted by surrogate methods, majorly of two groups: “on-grid” sampling and scattered sampling. The former is used in some methods based on polynomial approximations, e.g. stochastic collocation methods [2], and affected by the “curse of dimensionality” if the number of variables is large. The latter is more robust since it admits an arbitrary number of samples and arbitrary sample sites. This flexibility not only makes it tolerate sample failures (due to, e.g. poor convergence, as often observed in CFD models), but also makes an incorporation of pre-existing or additional samples possible and enables run-time adaptive sampling.

In this work we apply three gradient-employing surrogate methods, i.e. gradient-enhanced radial basis functions (GERBF), gradient-enhanced point-collocation polynomial chaos (GEPC) and gradient-enhanced Kriging (GEK) [13], and for the purpose of comparison, also the quasi-Monte Carlo quadrature, to a UQ test case where an RAE2822 airfoil is subject to random geometric perturbations, and we compare their efficiency in estimating some statistics and probability distribution of the resulting uncertain lift and drag coefficients. The number of CFD model

evaluations is kept small (≤ 200) in this numerical comparison to make it relevant to large-scale industrial applications.

2 Test Case

The test case we use in this work is a CFD model of the inviscid flow around a 2-dimensional RAE2822 airfoil at a Mach number of 0.73 and an angle of attack of 2.0 degrees. The source of uncertainty is the randomly perturbed airfoil geometry, i.e., the lower and upper surfaces of the airfoil's 2D section (as shown by the solid line in the right part of Figure 1) are each assumed to be subject to a Gaussian random perturbation in the direction normal to the surface. Let \mathbf{p}_l and \mathbf{p}_u denotes the original lower and upper surface respectively, the perturbed surfaces are

$$\mathbf{p}'_l = \mathbf{p}_l + \mathbf{n} \cdot \theta_l(x)$$

$$\mathbf{p}'_u = \mathbf{p}_u + \mathbf{n} \cdot \theta_u(x)$$

with $x \in [0, 1]$. \mathbf{n} is the local normal vector of the surface, $\theta(x)$ is a zero-mean Gaussian variable with standard deviations $\sigma(x)$, i.e.

$$\theta_l(x), \theta_u(x) \sim N(0, \sigma(x))$$

in which

$$\sigma(x) = 0.01 \cdot Z_{max} \cdot x(1-x) \cdot \beta(2,2)/1.5$$

with Z_{max} the maximum half-thickness of the airfoil, and β the Beta function. This setting makes the $\sigma(x)$ have its maximum (one percent of Z_{max}) at $x = 0.5$ and being zero at the two ends of the airfoil.

It is assumed that the random deformation is spatially correlated by a Gaussian type correlation function, i.e.

$$\begin{aligned} \text{cov}[\theta_l(x_1), \theta_l(x_2)] &= \sigma(x_1)\sigma(x_2) \exp\left(-\frac{(x_1-x_2)^2}{\ell^2}\right) \\ &= C(x_1, x_2) \end{aligned}$$

with $\ell = 0.2$. The same also applies to $\theta_u(x)$.

For the purpose of numerical computation, the correlated random fields $\theta_l(x)$ and $\theta_u(x)$ need to be represented in terms of uncorrelated random variables. This is furnished by Karhunen-Loève expansions (KLE) [1], e.g. for θ_l ,

$$\theta_l(x) = \sum_{i=1}^{\infty} \sqrt{\lambda_i} \xi_i \Phi_i(x)$$

where ξ_i are independent standard Gaussian variables. λ_i and $\Phi_i(x)$ are the eigenvalues and the eigenfunctions of $C(x_1, x_2)$, i.e., the solutions of the following integral equation,

$$\int_0^1 C(x_1, x_2) \Phi_i(x_1) dx_1 = \lambda_i \Phi_i(x_2) \quad \forall i = 1, 2, \dots$$

For practical problems the KLE needs to be truncated so that only a relatively small number of terms is kept, e.g. an approximation with κ terms:

$$\theta_l(x) \approx \sum_{i=1}^{\kappa} \sqrt{\lambda_i} \xi_i \Phi_i(x)$$

By taking $\kappa = 5$, $\theta_l(x)$ is parameterized by 5 independent standard Gaussian variables. Applying the same approximation to $\theta_u(x)$,

$$\theta_u(x) \approx \sum_{i=\kappa+1}^{2\kappa} \sqrt{\lambda_i} \xi_i \Phi_i(x)$$

we express the randomly perturbed airfoil surface as a function of 10 such variables. This KLE representation is optimal in the sense that it retains the original geometric variance to the maximum degree compared to any other linear-form representation with the same number of variables [11]. Figure 1 shows three examples of random perturbation in the upper and lower surface together with the corresponding perturbed RAE2822 airfoil geometry.

In this test case, the CFD model takes the input variables $\xi = \{\xi_1, \dots, \xi_{10}\}$ and yields the lift and drag coefficients, C_l and C_d , of the randomly perturbed airfoil. Hereafter, the model is denoted as $f(\xi)$ in this paper. We compare the efficiency of the candidate methods in estimating some target statistics, i.e. the means (μ_l, μ_d), the standard deviations (σ_l, σ_d) of C_l and C_d , and the exceedance probabilities $P_{l,j} =$

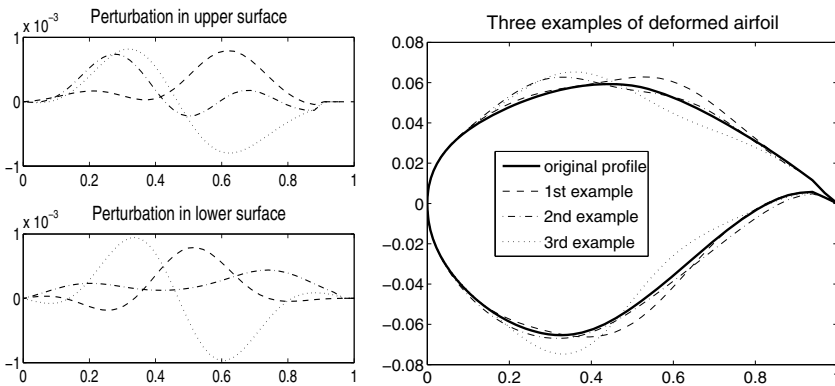


Fig. 1 Three examples of random perturbation in upper and lower surface (left) and three examples of the randomly perturbed airfoil geometry, with the perturbations ten-times exaggerated (right)

$\text{Pro}\{C_\ell < \mu_\ell - j \cdot \sigma_\ell\}$ and $P_{d,j} = \text{Pro}\{C_d > \mu_d + j \cdot \sigma_d\}$ with $j = 2, 3$. The accuracy of the statistics is judged by comparing with reference statistics obtained by a QMC integration with a large sample number ($N = 2 \times 10^5$).

3 Methods

Four methods are applied to the test case. They include three surrogate methods, i.e. gradient-enhanced radial basis functions (GERBF), gradient-enhanced Kriging (GEK) and gradient-enhanced point-collocation polynomial chaos (GEPC), and one direct integration method, i.e. quasi-Monte Carlo (QMC) quadrature. An introduction of them is made in this section.

Since the gradients of the SRQ with respect to all the ten variables are computed by an adjoint solver at an additional cost of approximately one evaluation of the CFD model, to account for this additional cost we introduce the term *elapsed time-penalized sample number* M by making $M = 2N$ for the three gradient-employing methods and $M = N$ for QMC, with N the number of evaluations of the CFD model. Compared to the cost of evaluating the CFD model the computational overhead of constructing surrogates is negligible, so in the efficiency comparison we use M as the measure of computational cost.

In the aspect of design of experiment, the study in [23] shows surrogate models based on samples with relatively high degree of uniformity (using Latin Hyper-cube sampling) are more accurate than those based on samples of lower degree of uniformity (using plain Monte-Carlo sampling). For all the four methods in this work we adopt the QMC sampling scheme [7] because it achieves even higher degree of sample uniformity than Latin Hyper-cube sampling.

We use the DLR-TAU code [10] to solve the CFD model. The geometry perturbation is implemented by using a mesh deformation tool based on radial basis functions incorporated in the DLR-TAU code as described in [14].

3.1 Quasi-Monte Carlo Quadrature

Quasi-Monte Carlo (QMC) quadrature [7] samples at a *low discrepancy* set of points generated by deterministic number-theoretic formulas. The “discrepancy” here is a measurement of how much the distribution of this set of points deviates from the underlying pdf. A low discrepancy set of points achieves a higher degree of uniformity with respect to a given pdf than a pseudo-random set of points does. So QMC is usually much more efficient than a Monte Carlo quadrature. The error bound of QMC is of order $O(N^{-1}(\log N)^d)$ in which d is the number of variables. In many cases this is quite a loose upper bound of the error, i.e. QMC often performs better than that.

A variety of low discrepancy point sets exist, e.g. Van der Corput, Halton, Sobol, Hammersley and Niederreiter point set. The last one is used in this work as it is considered the most efficient when d is large [19]. The point set is generated by the program from [4]. The statistics of the SRQ are directly computed from the samples.

3.2 Gradient-Enhanced Radial Basis Functions

The radial basis function (RBF) method [6] approximates an unknown function by a weighted linear combination of *radial basis functions* each being radially symmetric about a center. An RBF approximation takes the form

$$\hat{f}(\boldsymbol{\xi}) = \sum_{i=1}^N w_i \phi_i(\|\boldsymbol{\xi} - \boldsymbol{\xi}^{(i)}\|)$$

where ϕ_i are radial basis functions, $\|\cdot\|$ denotes the Euclidean norm, and $\boldsymbol{\xi}^{(i)}$ are the N sample points each taken as the center of a radial basis function. Making $\hat{f}(\boldsymbol{\xi})$ interpolate the N samples leads to N linear equations. The coefficients w_i are determined by solving this linear system.

Denoting the Euclidean distance from the center as r , popular types of $\phi(r)$ include $\sqrt{r^2 + a^2}$ (multiquadric), $1/\sqrt{r^2 + a^2}$ (inverse multiquadric), $\exp(-a^2 r^2)$ (Gaussian) and $r^2 \ln(ar)$ (thin plate spline), in which a is a parameter to be fine-tuned for a particular set of samples. Gradient-employing versions of RBF were proposed in [11, 20] where first-order derivatives of the SRQ are exploited and second-order derivatives of RBF are involved in the system.

In this work we propose a different gradient-employing RBF method that involves only the first-order derivative of RBF, termed gradient-enhanced RBF (GERBF). To accommodate the gradient informations of the SRQ, this method introduces additional RBF that are centered at *non-sampled* points, i.e. an GERBF approximation is

$$\hat{f}(\boldsymbol{\xi}) = \sum_{i=1}^K w_i \phi_i(\|\boldsymbol{\xi} - \boldsymbol{\xi}^{(i)}\|), \quad \text{with } N < K \leq N(1+d)$$

The $\boldsymbol{\xi}^{(i)}$ with $i \leq N$ are sampled points, those with $i > N$ are non-sampled points which can be chosen randomly as long as none of them duplicates the sampled ones. The coefficients $\mathbf{w} = \{w_0, w_1, \dots, w_K\}^T$ are determined by solving the following system,

$$\Phi \mathbf{w} = \mathbf{f}$$

in which

$$\Phi = \begin{bmatrix} \Phi_1(\xi^{(1)}) & \Phi_2(\xi^{(1)}) & \dots & \Phi_K(\xi^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1(\xi^{(N)}) & \Phi_2(\xi^{(N)}) & \dots & \Phi_K(\xi^{(N)}) \\ \Phi_1^{(1)}(\xi^{(1)}) & \Phi_2^{(1)}(\xi^{(1)}) & \dots & \Phi_K^{(1)}(\xi^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1^{(1)}(\xi^{(N)}) & \Phi_2^{(1)}(\xi^{(N)}) & \dots & \Phi_K^{(1)}(\xi^{(N)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1^{(d)}(\xi^{(1)}) & \Phi_2^{(d)}(\xi^{(1)}) & \dots & \Phi_K^{(d)}(\xi^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1^{(d)}(\xi^{(N)}) & \Phi_2^{(d)}(\xi^{(N)}) & \dots & \Phi_K^{(d)}(\xi^{(N)}) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(\xi^{(1)}) \\ \vdots \\ f(\xi^{(N)}) \\ f^{(1)}(\xi^{(1)}) \\ \vdots \\ f^{(1)}(\xi^{(N)}) \\ \vdots \\ f^{(d)}(\xi^{(1)}) \\ \vdots \\ f^{(d)}(\xi^{(N)}) \end{bmatrix}$$

with $\Phi_i^{(j)} = \partial \Phi_i / \partial \xi_j$, $f^{(j)} = \partial f / \partial \xi_j$. We chose $K = \frac{N}{2}(1 + d)$ in this work, which results in an over-determined system that is solved by a Least Squares method.

A numerical comparison of the accuracy of the aforementioned four types of RBF in approximating this CFD model $f(\xi)$ was made by the author. The result favors the inverse multiquadric RBF which is therefore used in this work for the comparison with other UQ methods. The internal parameter a is fine-tuned by a *leave-one-out* error minimizing procedure as in [3].

For this UQ job we first establish a GERBF surrogate model of $f(\xi)$ based on QMC samples of the CFD model, and integrate for the target statistics and pdf by a large number (10^5) of QMC samples on the surrogate model.

3.3 Gradient-Enhanced Kriging Method

Kriging [9] approximates $f(\xi)$ by a weighted linear combination of samples, i.e.

$$\hat{f}(\xi) = \gamma(\xi) + \sum_{i=1}^N w_i(\xi^{(i)}) f(\xi^{(i)})$$

where $f(\xi^{(i)})$ are N samples of the SRQ. γ and w_i are determined by minimizing the variance of the error $\epsilon = f - \hat{f}$ with the assumptions that the expectation of ϵ is zero and that $f(\xi)$ honors a spatial correlation model. We use direct gradient-enhanced Kriging (GEK) [8] that incorporates gradient information as secondary samples by an extended spatial correlation model that accommodates gradients. We implement GEK using the *Surrogate-Modeling for Aero-Data Toolbox (SMART)* [13] developed at DLR, opting for ordinary Kriging and a correlation model of spline type which is considered the most efficient in similar situations in [17]. The internal parameters of the correlation model are fine-tuned to fit the sampled data by a maximum likelihood estimation [24].

For this UQ job we first establish a GEK surrogate model of $f(\boldsymbol{\xi})$ based on QMC samples of the CFD model, and integrate for the target statistics and pdf by a large number (10^5) of QMC samples on the surrogate model.

3.4 Gradient-Enhanced Point-Collocation Polynomial Chaos Method

According to Wiener [22], $f(\boldsymbol{\xi})$ can be approximated by a truncated *polynomial chaos expansion* (PCE)

$$\hat{f}(\boldsymbol{\xi}) = \sum_{i=0}^K c_i \Psi_i(\boldsymbol{\xi}) \tag{1}$$

where Ψ_i is Hermite polynomial chaos (PC) to which a detailed description can be found in, e.g. [18]. The total number of terms is $K = (p + d)! / (p! d!)$ with p the order of PC.

To determine the coefficients c_i we use a point-collocation method similar to the one used in [15], the difference being that we utilize gradient information. In this gradient-enhanced point-collocation polynomial chaos (GEPC) method the $\mathbf{c} = \{c_0, c_1, \dots, c_K\}^T$ is determined by solving the following system,

$$\boldsymbol{\Psi} \mathbf{c} = \mathbf{f}$$

$$\text{with } \boldsymbol{\Psi} = \begin{bmatrix} \Psi_0(\boldsymbol{\xi}^{(1)}) & \Psi_1(\boldsymbol{\xi}^{(1)}) & \dots & \Psi_K(\boldsymbol{\xi}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0(\boldsymbol{\xi}^{(N)}) & \Psi_1(\boldsymbol{\xi}^{(N)}) & \dots & \Psi_K(\boldsymbol{\xi}^{(N)}) \\ \Psi_0^{(1)}(\boldsymbol{\xi}^{(1)}) & \Psi_1^{(1)}(\boldsymbol{\xi}^{(1)}) & \dots & \Psi_K^{(1)}(\boldsymbol{\xi}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0^{(1)}(\boldsymbol{\xi}^{(N)}) & \Psi_1^{(1)}(\boldsymbol{\xi}^{(N)}) & \dots & \Psi_K^{(1)}(\boldsymbol{\xi}^{(N)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0^{(d)}(\boldsymbol{\xi}^{(1)}) & \Psi_1^{(d)}(\boldsymbol{\xi}^{(1)}) & \dots & \Psi_K^{(d)}(\boldsymbol{\xi}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi_0^{(d)}(\boldsymbol{\xi}^{(N)}) & \Psi_1^{(d)}(\boldsymbol{\xi}^{(N)}) & \dots & \Psi_K^{(d)}(\boldsymbol{\xi}^{(N)}) \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f(\boldsymbol{\xi}^{(1)}) \\ \vdots \\ f(\boldsymbol{\xi}^{(N)}) \\ f^{(1)}(\boldsymbol{\xi}^{(1)}) \\ \vdots \\ f^{(1)}(\boldsymbol{\xi}^{(N)}) \\ \vdots \\ f^{(d)}(\boldsymbol{\xi}^{(1)}) \\ \vdots \\ f^{(d)}(\boldsymbol{\xi}^{(N)}) \end{bmatrix}$$

where $\Psi_i^{(k)} = \partial \Psi_i / \partial \xi_k$, $f^{(k)} = \partial f / \partial \xi_k$, and $\boldsymbol{\xi}^{(i)} = \{\xi_1, \xi_2, \dots, \xi_d\}_i$ with $i = 1, \dots, N$ denote the sample points. The K is chosen to be half of the number of available ‘‘conditions’’, $N(1 + d)$, for the best performance according to [15]. This over-determined system is solved by a Least Squares method.

For this UQ job we first establish a GEPC surrogate model of $f(\boldsymbol{\xi})$ based on QMC samples of the CFD model, and compute the mean and the variance of $f(\boldsymbol{\xi})$ directly from the coefficients,

$$\mu = c_0, \quad \sigma^2 = \sum_{i=1}^K (c_i)^2 \cdot \mathbb{E}[\Psi_i^2(\boldsymbol{\xi})] \tag{2}$$

The exceedance probabilities and pdf are integrated by a large number (10^5) of QMC samples on the surrogate model.

4 Results and Discussion

The results of the efficiency comparison are shown in Figure 2 to 5. Figure 2 and 3 show the errors of the four methods in estimating the target statistics of C_ℓ and C_d . It is observed there that the three gradient-employing surrogate methods, GEK, GEPC and GERBF are more efficient than the QMC method since the former three reduce their errors faster with an increasing cost measure M . Figure 4 and 5 depict the estimated pdf of C_ℓ and C_d obtained by the four methods, comparing with the reference pdf. There we see that for the same computational cost, the surrogate methods yield much more accurate pdf's. This is consistent with their relative performance in estimating the statistics.

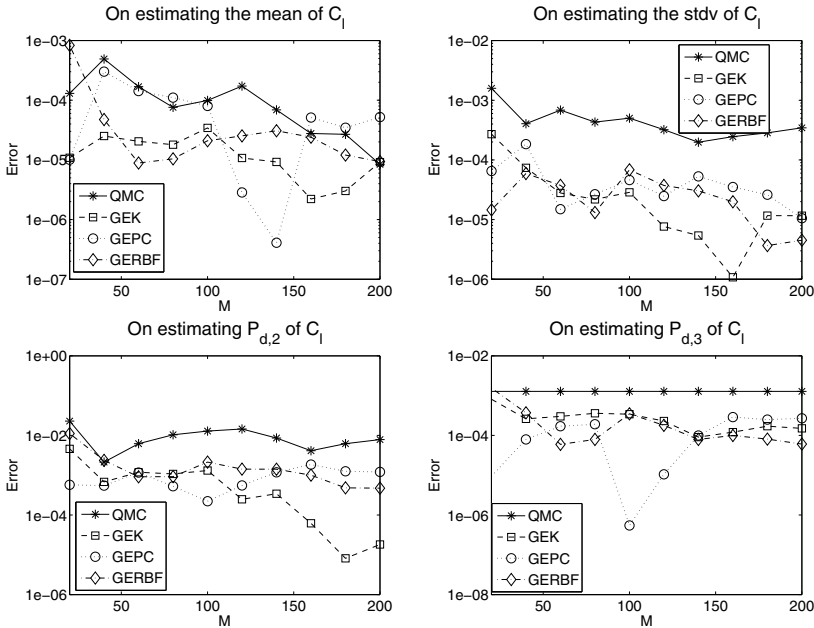


Fig. 2 Error in estimating mean, standard deviation (upper row) and exceedance probabilities (lower row) of C_ℓ

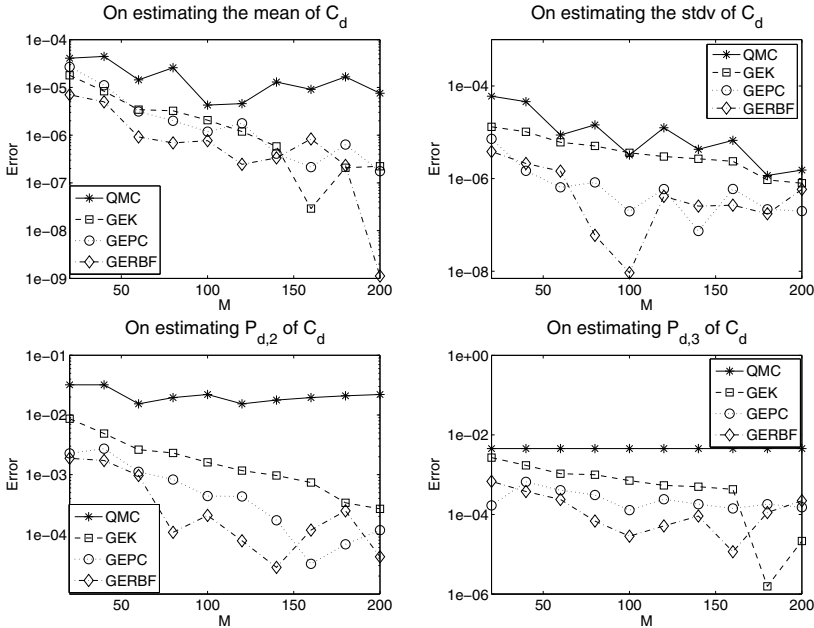


Fig. 3 Error in estimating mean, standard deviation (upper row) and exceedance probabilities (lower row) of C_d

One of the reasons for the relatively better performance of the surrogate methods is that they utilize more information with the same computational cost M , i.e. they use $(1 + d)\frac{M}{2}$ conditions (SRQ samples and gradients) while a direct integration method like QMC uses M conditions (SRQ samples only). This advantage is due to the cheaper cost of obtaining gradients by an adjoint solver in the case that the number of SRQ's is smaller than the number of variables d , and is expected to increase with an increasing number of variables, d .

Although it seems that GEK and GERBF perform better than the other surrogate methods in estimating statistics of C_ℓ and C_d respectively, it may not be appropriate to base a general conclusion on that. In Figure 4 and 5 we see the three surrogate methods have similar accuracy in their estimated pdf of C_ℓ and C_d .

The efficiency of GEK or GERBF is sensitive to the choice of the internal covariance model or the radial basis function and also to the value of the internal parameters, and excellent configurations of them are problem- and data-dependent. In this work, different techniques are used for the optimization of the internal parameters, i.e., maximum likelihood optimization for GEK and leave-one-out error minimization for GERBF. This may also influence their relative efficiency, possibly differently in the C_ℓ and C_d cases. Due to the complex nature of comparative efficiency of

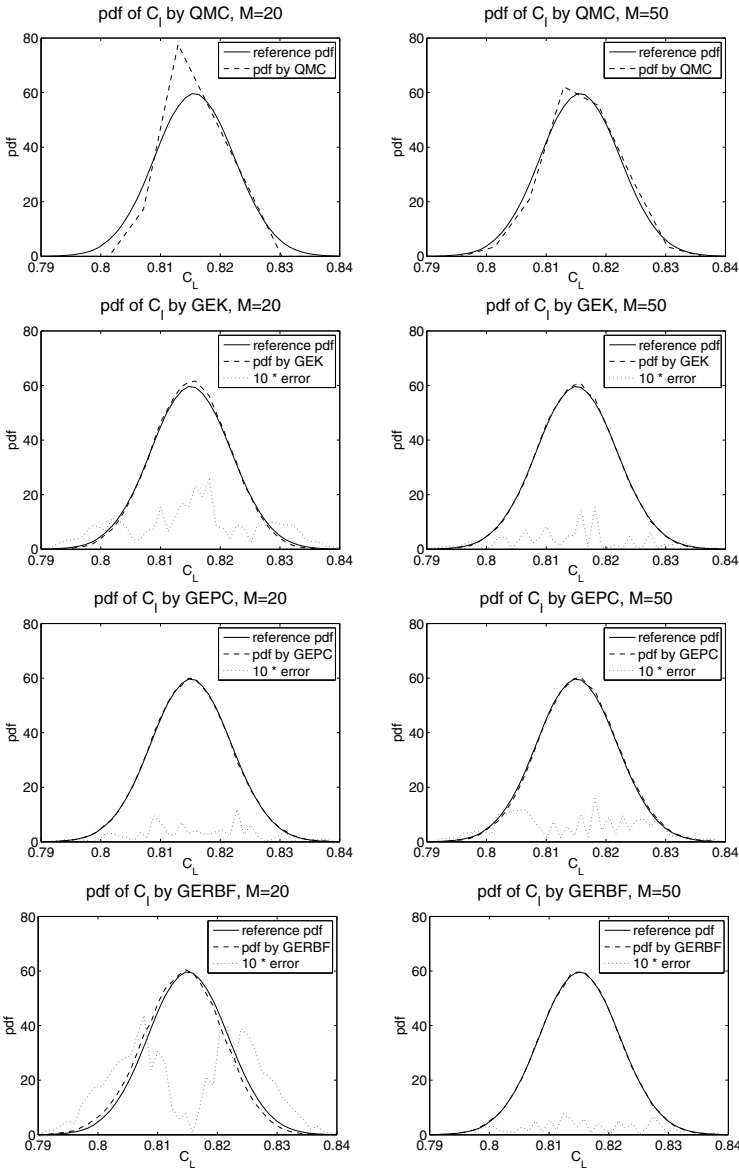


Fig. 4 Estimated pdf (in dash line) of C_ℓ by QMC (1st row), GEK (2nd row), GEPC (3rd row) and GERBF(4th row) at $M = 20$ (left) and $M = 50$ (right), dotted line shows the 10-times scaled up error of the estimated pdf

surrogate methods with different configurations and internal optimization techniques and different target SRQ, here we do not try to draw a conclusion on this issue.

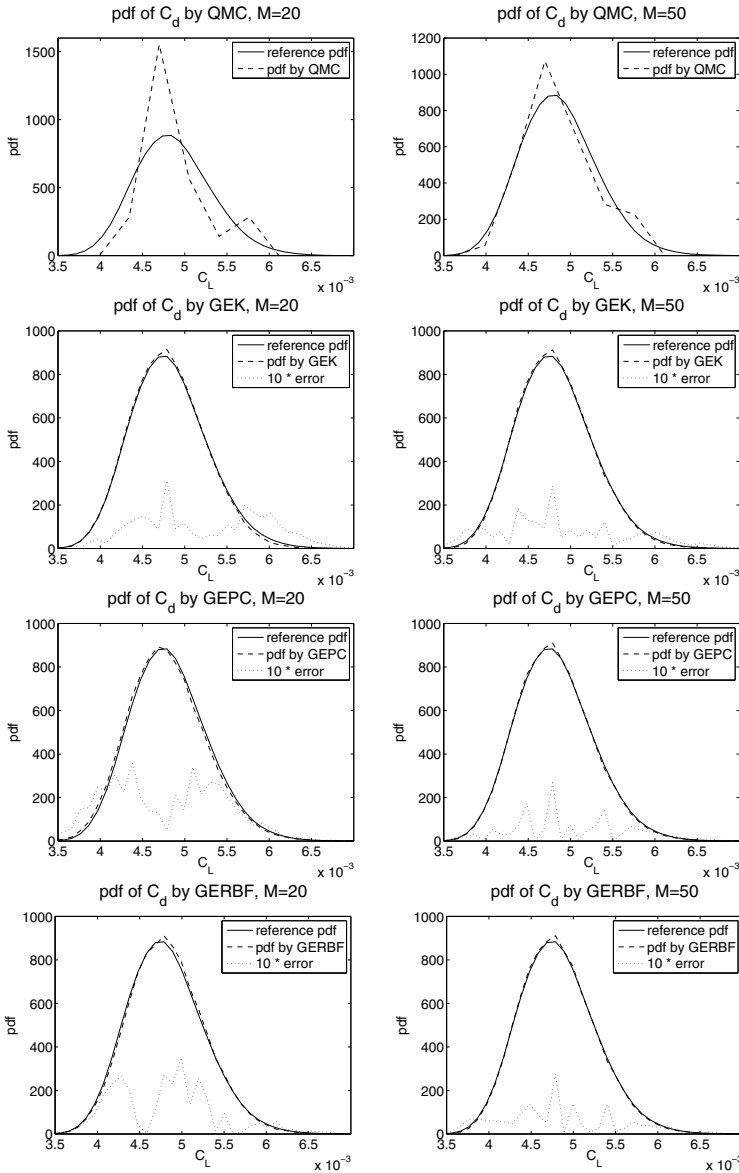


Fig. 5 Estimated pdf (in dash line) of C_d by QMC (1st row), GEK (2nd row), GEPC (3rd row) and GERBF(4th row) at $M = 20$ (left) and $M = 50$ (right), dotted line shows the 10-times scaled up error of the estimated pdf

In the estimation of the statistics of C_ℓ we see GEPC is not always reducing its error with an increasing M . This might be ascribed to the fact that the number of polynomial chaos (PC) terms is not truncated according to the order of PC, but to

an arbitrary number which is half of the number of available conditions. A set of PC terms that is “incomplete” for a particular order might not lead to more accurate approximations than a set with less number of terms but “complete” for a lower order. Nevertheless, GEPC has a favored property that we have no burden of choosing the best-fitting configuration for it.

5 Summary

Gradient-employing surrogate methods have an advantage in handling aerodynamic uncertainty quantification (UQ) problems in the cases that an adjoint solver is used and the number of system response quantities (SRQ) is smaller than the number of variables so that the gradients of SRQ can be obtained at a reduced cost. These methods construct surrogates of the CFD model so that the statistics of an uncertain SRQ can be integrated on the surrogates models.

For investigating the efficiency of the different UQ methods we set up a test case where the geometry of an RAE2822 airfoil is perturbed by a Gaussian random field which is parameterized by 10 independent variables. Three surrogate methods, gradient-enhanced radial basis functions, gradient-enhanced point-collocation polynomial chaos and gradient-enhanced Kriging, together with a direct integration method, quasi-Monte Carlo (QMC) quadrature, are applied to the test case and compared in their efficiency in estimating some statistics and probability distribution of the uncertain lift and drag coefficients. The results show that with the same computational effort the gradient-employing surrogate methods achieve better accuracy than the QMC does.

References

- [1] Adler, R.J., Taylor, J.E.: *Random Fields and Geometry*. Springer, Berlin (2007)
- [2] Babuška, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Numer. Anal.* 45, 1005–1034 (2007)
- [3] Bompard, M., Peter, J., Désidéri, J.A.: Surrogate models based on function and derivative values for aerodynamic global optimization. In: *V European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2010, Lisbon, Portugal* (2010)
- [4] Bratley, P., Fox, B.L., Niederreiter, H.: Programs to generate Niederreiter’s low-discrepancy sequences. *ACM Trans. Math. Softw.* 20(4), 494–495 (1994)
- [5] Brezillon, J., Dwight, R.: Discrete adjoint of the navier-stokes equations for aerodynamic shape optimization. In: *EUROGEN 2005 - Sixth Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Munich, Germany* (2005)
- [6] Buhmann, M.D.: Radial basis functions. *Acta Numerica* 9, 1–38 (2000)
- [7] Caffisch, R.E.: Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica* 7, 1–49 (1998)

- [8] Chung, H.S., Alonso, J.J.: Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. AIAA paper 2002–0317 (2002)
- [9] Cressie, N.: *Statistics for Spatial Data*. Wiley, New York (1991)
- [10] Gerhold, T., Friedrich, O., Evans, J., Galle, M.: Calculation of complex three-dimensional configurations employing the DLR-TAU-Code. AIAA paper 97-0167 (1997)
- [11] Giannakoglou, K.C., Papadimitriou, D.I., Kambolis, I.C.: Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Computer Methods in Applied Mechanics and Engineering* 195(44-47), 6312–6329 (2006)
- [12] Giunta, A.A., Eldred, M.S., Castro, J.P.: Uncertainty quantification using response surface approximation. In: 9th ASCE Specialty Conference on Probabilistic Mechanics and Structural Reliability, Albuquerque, New Mexico, USA (2004)
- [13] Han, Z.H., Görtz, S., Zimmermann, R.: Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Journal of Aerospace Science and Technology* (2012)
- [14] Heinrich, R., Kroll, N., Neumann, J., Nagel, B.: Fluid-structure coupling for aerodynamic analysis and design: a DLR perspective. AIAA paper 2008–561 (2008)
- [15] Hosder, S., Walters, R.W., Balch, M.: Efficient sampling for non-intrusive polynomial chaos applications with multiple uncertain input variables. AIAA paper 2007–1939 (2007)
- [16] Litvinenko, A., Matthies, H.G.: Sparse data formats and efficient numerical methods for uncertainties quantification in numerical aerodynamics. In: ECCM IV: Solids, Structures and Coupled Problems in Engineering (2010)
- [17] Liu, D.S.: A Best Practice Guide: Efficient Quantification of Aerodynamic Uncertainties. Tech. rep., DLR, IB 124-2012/2 (2012)
- [18] Matthies, H.G.: Uncertainty quantification with stochastic finite elements. In: Stein, E., de Borst, R., Hughes, T.R.J. (eds.) *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Chichester (2007)
- [19] Niederreiter, H.: *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia (1992)
- [20] Ong, Y., Lum, K., Nair, P.: Hybrid evolutionary algorithm with hermite radial basis function interpolants for computationally expensive adjoint solvers. *Computational Optimization and Applications* 39, 97–119 (2008)
- [21] Schillings, C., Schulz, V.: On the influence of robustness measures on shape optimization with stochastic uncertainties. Tech. rep., Trierer Forschungsbericht 12-2, Universität Trier (2012)
- [22] Wiener, N.: The homogeneous chaos. *Amer. J. Math.* 60, 897–936 (1938)
- [23] Zhao, D., Xue, D.: A comparative study of metamodeling methods considering sample quality merits. *Struc. Multidisc. Optim.* (42), 923–938 (2010)
- [24] Zimmermann, R.: Asymptotic behavior of the likelihood function of covariance matrices of spatial gaussian processes. *J. App. Math.*, Article ID 494070 (2010)

Optimal Aerodynamic Design under Uncertainty

Volker Schulz and Claudia Schillings

Abstract. Recently, optimization has become an integral part of the aerodynamic design process chain. However, because of uncertainties with respect to the flight conditions and geometry uncertainties, a design optimized by a traditional design optimization method seeking only optimality may not achieve its expected performance. Robust optimization deals with optimal designs, which are robust with respect to small (or even large) perturbations of the optimization setpoint conditions. That means, the optimal designs computed should still be good designs, even if the input parameters for the optimization problem formulation are changed by a non-negligible amount. Thus even more experimental or numerical effort can be saved. In this paper, we aim at an improvement of existing simulation and optimization technology, developed in the German collaborative effort MEGADESIGN¹, so that numerical uncertainties are identified, quantized and included in the overall optimization procedure, thus making robust design in this sense possible. We introduce two robust formulations of the aerodynamic optimization problem which we numerically compare in a 2d testcase under uncertain flight conditions. Beside the scalar valued uncertainties we consider the shape itself as an uncertainty source and apply a Karhunen-Loève expansion to approximate the infinite-dimensional probability space. To overcome the curse of dimensionality an adaptively refined sparse grid is used in order to compute statistics of the solution.

Volker Schulz · Claudia Schillings

FB IV Department of Mathematics, University of Trier, Universitätsring 15,
54296 Trier, Germany

e-mail: {volker.schulz,claudia.schillings}@uni-trier.de

¹ MEGADESIGN Aerodynamic simulation and optimization in aircraft design, German national project funded by the German Federal Ministry of Economics and Labour (BMWA).

1 Introduction

Uncertainties pose problems for the reliability of numerical computations and their results in all technical contexts one can think of. They have the potential to render worthless even highly sophisticated numerical approaches, since their conclusions do not realize in practice due to unavoidable variations in problem data. The proper treatment of these uncertainties within a numerical context is a very important challenge. This paper is devoted to the enhancement of highly efficient optimal design techniques developed in the framework of MEGADESIGN by a robustness component, which tries to make the optimal design generated a still good design, if the setting of a specific design point is varied.

2 Aleatory Uncertainties in Aerodynamic Design

Aleatory uncertainties arises because of natural, unpredictable variations of the boundary conditions. Additional knowledge cannot reduce aleatory uncertainties, but it may be useful in getting a better characterization of the variability. In order to formulate the robust design optimization problem, we analyze the boundary conditions and input parameters identifying the uncertainties which cannot be avoided at all before constructing an aircraft [39].

In the following, we distinguish two types of uncertainties: uncertainties with respect to the flight conditions and geometrical uncertainties.

The main characteristics of the macroscopic flight conditions are angle of incidence, the velocity (Mach number) of the plane, the density of air and the Reynolds number. The uncertainty of these parameters mostly results from atmospheric turbulences which can occur during a flight. Gusts causes changes of the velocity in the range of $\pm 10 \frac{m}{s}$. Measurements of the changes in the angle of attack and the density are not published so far, so they are assumed to be less than 10% of the setpoint. The variations of the Reynolds number will only effect the simulation, if the Reynolds number is in the range of $12 - 15 \cdot 10^6$, that means this uncertain parameter has not to be taken into account in our testcases.

On the other hand, we consider the shape itself as an uncertainty source. The real shape may vary from the planned shape due to manufacturing tolerances, temporary factors like icing e.g. or fatigue of material. Since there are so many factors having effects on the shape, this uncertainty has to be considered in the optimization problem in order to produce a design which is robust to small perturbations of the shape itself. In the literature, there can be found only a few papers on this topic investigating the influence of variations of the profile (cf. [17], [32]).

2.1 Mathematical Description of the Uncertainties

Since we want to avoid a parametrization of the uncertainties which would lead to a reduction of the space of realizations, we choose a stochastic approach in order to include the uncertainties in the optimization problem. Furthermore, this approach allows to adapt the robust optimization to new information of the uncertain parameter, e.g. if new measurements are available, so that a general framework of robust aerodynamic design can be developed.

The proper treatment of the uncertainties within a numerical context is a very important challenge, since the simulation and also optimization under uncertainties is a fast growing field of research. Again, we distinguish between the uncertainties with respect to the flight conditions, the scalar-valued uncertainties, and the geometrical uncertainties, the function-valued uncertainties.

2.1.1 Scalar-Valued Uncertainties

The scalar-valued uncertainties, e.g. the Mach number, are modeled as real-valued, continuous random variables

$$s : \Omega \rightarrow \mathbb{R}, \quad (1)$$

defined on a given probability space (Ω, Y, P) . They are characterized by a given probability density function

$$\varphi_{truncated} : \mathbb{R} \rightarrow \mathbb{R}_+. \quad (2)$$

We assume (mainly due to lack of statistical data) a truncated normal distribution of the perturbations ensuring that the realizations lie in between the given bounds. Furthermore, the mean value of the random variable corresponds with the value of the deterministic model. These assumptions are widely used in order to describe uncertainties in CFD (cf. [32]). Nevertheless, the model need to be adapted to measurements, if available.

2.1.2 Function-Valued Uncertainties

The geometrical uncertainties also depend on the geometry itself, so they are modeled as a Gaussian random field

$$\psi : \Gamma \times \Omega \rightarrow \mathbb{R}, \quad (3)$$

defined on a probability space (Ω, Y, P) and on the shape of the airfoil Γ . In each point x of the shape Γ , the uncertainty is described by a normally distributed random variable $\psi(x, \cdot) : \Omega \rightarrow \mathbb{R}$. Additionally, the second order statistics, the mean value and the covariance function, are given to fully describe the random field. According to the scalar-valued uncertainties, the mean value of the random field ψ is equal to

0, since we expect no perturbations and the squared exponential covariance function describes the interaction between the random variables on the shape:

$$\mathbb{E}(\boldsymbol{\psi}(x, \zeta)) = \boldsymbol{\psi}_0(x) = \mathbf{0} \quad \forall x \in \Gamma \quad (4)$$

$$\text{Cov}(x, y) = b^2 \cdot \exp\left(-\frac{\|x-y\|^2}{l^2}\right) \quad \forall x, y \in \Gamma \quad (5)$$

The parameter l determines how quickly the covariance falls off and b controls the magnitude of the bumps. A squared exponential covariance function is chosen, since the resulting perturbed geometry is smooth due to the smoothness of the random field.

Then, a perturbed geometry is given as

$$v(x, \zeta) = x + \boldsymbol{\psi}(x, \zeta) \cdot \mathbf{n}(x) \quad \forall x \in \Gamma, \zeta \in \Omega \quad (6)$$

where \mathbf{n} is the unit vector in x normal to the profile Γ . As we need to compute statistics of the flow depending on the uncertainty in our optimization algorithm, we have to approximate and discretize the probability spaces. In the next chapter, we will introduce the Karhunen-Loève-Expansion which provides an approximation of the random field $\boldsymbol{\psi}$ for the numerical evaluation of such statistics and efficient discretization techniques of the probability space.

2.2 Karhunen-Loève-Expansion

The Karhunen-Loève-Expansion, also known as Proper Orthogonal Decomposition, represents the random field as a infinite linear combination of orthogonal functions chosen as the eigenfunctions of the covariance function [23], [31]. The Karhunen-Loève-Expansion of the Gaussian random field $\boldsymbol{\psi}$ is given as:

$$\boldsymbol{\psi}(x, \zeta) = \boldsymbol{\psi}_0(x) + \sum_{i=1}^{\infty} \sqrt{\lambda_i} z_i(x) Y_i(\zeta) \quad (7)$$

$$= \sum_{i=1}^{\infty} \sqrt{\lambda_i} z_i(x) Y_i(\zeta) \quad x \in \Gamma, \zeta \in \Omega \quad (8)$$

where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_i \geq \dots \geq 0$ and z_i are the eigenvalues and eigenfunctions of the covariance function Cov which is symmetric and positive definite by definition. The deterministic eigenfunctions z_i are obtained from the spectral decomposition of the covariance function via solution of

$$\int_{\Gamma} \text{Cov}(x, y) z_i(y) dy = \lambda_i z_i(x). \quad (9)$$

Having the eigenpairs, the uncorrelated Gaussian random variables Y_i in equation (8) can be expressed as

$$Y_i(\zeta) = \frac{1}{\sqrt{\lambda_i}} \int_{\Gamma} \psi(x, \zeta) z_i(x) dx \quad j = 1, 2, \dots \quad (10)$$

with zero mean and unit variance, i.e. $\mathbb{E}(Y_i) = 0$ and $\mathbb{E}(Y_i Y_j) = \delta_{ij}$ for $j = 1, 2, \dots$ [5]. In the special case of a Gaussian random field, uncorrelated random variables are independent as well, which is an important property we will need later on for the sparse grid.

Truncating now the Karhunen-Loève-Expansion after a finite number of terms, we obtain the approximation of the random field ψ

$$\psi_d(x, \zeta) = \sum_{i=1}^d \sqrt{\lambda_i} z_i(x) Y_i(\zeta) \quad x \in \Gamma, \zeta \in \Omega. \quad (11)$$

The corresponding covariance function is given by

$$\text{Cov}_d(x, y) = \sum_{i=1}^d \lambda_i z_i(x) z_i(y). \quad (12)$$

In [15], it is shown that the eigenfunction basis $\{z_i\}$ is optimal in the sense that the mean square error resulting from the truncation after the d th term is minimized.

The following approximation error representation is then obtained by Mercer's theorem [34]

$$\lim_{d \rightarrow \infty} \left\{ \sup_{\Gamma} \int_{\Omega} (\psi - \psi_d)^2 d\mathcal{P}(\zeta) \right\} = \lim_{d \rightarrow \infty} \left\{ \sup_{\Gamma} \left(\sum_{j=d+1}^{\infty} \lambda_j z_j^2 \right) \right\} = 0. \quad (13)$$

So, ψ_d may provide a suitable approximation of ψ , if the eigenvalues decay sufficiently fast and d is large enough [5]. If one assumes a Gaussian covariance function (cf. (5)), the eigenvalues will exponentially decay towards zero. The proof of this behaviour of the eigenvalues can be found e.g. [11]. This paper also provides a fast algorithm based on a kernel independent fast multipole method to compute the Karhunen-Loève approximation. Another approach to solve the large eigenvalue problem arising from the Karhunen-Loève-Expansion can be found in [25]. They introduce a Krylov subspace method with a sparse matrix approximation using sparse hierarchical matrix techniques to solve it.

3 Robust Shape Optimization Problem

The usual single setpoint aerodynamic shape optimization problem can be described in the following rather abstract form

$$\min_{y,p} f(y,p) \tag{14}$$

$$\text{s.t. } c(y,p) = 0 \tag{15}$$

$$h(y,p) \geq 0 \tag{16}$$

We think of equation (15) as the discretized outer flow equation around, e.g., an airfoil described by geometry parameter $p \in \mathbb{R}^{n_p}$. The vector y is the state vector (velocities, pressure,...) of the flow model (15) and we assume that (15) can be solved uniquely for y for all reasonable geometries p . The objective in (14) $f : (y,p) \mapsto f(y,p) \in \mathbb{R}$ typically is the drag to be minimized. The restriction (16) typically denotes lift or pitching moment requirements. To make the discussion here simpler, we assume a scalar valued restriction, i.e., $h(y,p) \in \mathbb{R}$. The generalization of the discussions below to more than one restriction is straight forward. In contrast to previous papers on robust aerodynamic optimization, we treat the angle of attack as an fixed parameter which is not adjusted to reach the required lift (cf. e.g. Ref. [29], Ref. [21], Ref. [36]).

The general deterministic problem formulation (14,16) is influenced by stochastic perturbations. We assume that there are uncertain disturbances involved in the form of real-valued random variables $s : \Omega \rightarrow \mathbb{R}$ (or random vectors) associated with a probability measure \mathcal{P} with Lebesgue density $\varphi : \mathbb{R} \rightarrow \mathbb{R}_0^+$ such that the expected value of s can be written as

$$E(s) = \int_{\Omega} s(\zeta) d\mathcal{P}(\zeta) = \int_{\mathbb{R}} x\varphi(x) dx$$

and the expected value of any function $g : \mathbb{R} \rightarrow \mathbb{R}$ is written as

$$E(g(s)) = \int_{\Omega} g(s(\zeta)) d\mathcal{P}(\zeta) = \int_{\mathbb{R}} g(x)\varphi(x) dx$$

The dependence can arise in all aspects, i.e., a naive stochastic variant might be rewritten as

$$\min_{y,p} f(y,p,s) \tag{17}$$

$$\text{s.t. } c(y,p,s) = 0 \tag{18}$$

$$h(y,p,s) \geq 0 \tag{19}$$

This formulation still treats the uncertain parameter as an additional fixed parameter. The optimal solution should be stable with respect to stochastic variations in s . The literature can be classified in the following ideal classes: min-max formulation, semi-infinite formulation and chance constraints.

3.1 Min-Max Formulations

The min-max formulation aims at the worst-case scenario.

$$\min_{y,p} \max_{\zeta \in \Omega} f(y, p, s(\zeta)) \quad (20)$$

$$\text{s.t. } c(y, p, s(\zeta)) = 0, \forall \zeta \in \Omega \quad (21)$$

$$h(y, p, s(\zeta)) \geq 0, \forall \zeta \in \Omega \quad (22)$$

The min-max formulation is obviously independent of the stochastic measure \mathcal{P} and thus needs only the realizations of the random variable s as input. If the probability density function of the uncertain parameter is not available, this approach could potentially be an attractive strategy. Otherwise, this formulaion will ignore problem specific information, if it is at hand and will lead to overly conservative designs. We do not treat this formulation furthermore in this paper.

3.2 Semi-infinite Formulations

The semi-infinite reformulation aims at optimizing the average objective function but maintaining the feasibility with respect to the constraints everywhere. Thus, it aims at an average optimal and always feasible robust solution. The ideal formulation is of the form

$$\min_{y,p} \int_{\Omega} f(y, p, s(\zeta)) d\mathcal{P}(\zeta) \quad (23)$$

$$\text{s.t. } c(y, p, s(\zeta)) = 0, \forall \zeta \in \Omega \quad (24)$$

$$h(y, p, s(\zeta)) \geq 0, \forall \zeta \in \Omega \quad (25)$$

This definition of robustness can also be found in Ref.[21] and in Ref.[30]. Semi-infinite optimization problems have been treated directly so far only for rather small and weakly nonlinear problems, e.g. Ref.[10]. For the numerical treatment of complicated design tasks, one has to approximate the integral in the objective (23). Considering scalar-valued uncertainties, we assume a truncated normal distribution, that means in the multivariate case $s \sim \frac{1}{const} N(v, C) \cdot \mathbf{1}_R$, $R \subset \mathbb{R}^d$ with expected value

$$\text{vector } v, \text{ Covariance } C \text{ and indicator function } \mathbf{1}_R(x) = \begin{cases} 1, & \text{if } x \in R \\ 0, & \text{if } x \notin R \end{cases}$$

The integral in (23) can be efficiently evaluated by a Gaussian quadrature for small stochastic dimensions, where the quadrature points $\{s_i\}_{i=1}^N$ are the roots of a polynomial belonging to a class of orthogonal polynomials. Due to the exponential growth of the effort with increasing dimension, the full tensor product Gaussian quadrature rule should be replaced in the higher dimensional case by Smolyak type

algorithms which use a recursive contribution of lower-order tensor products to estimate the integral. We will discuss this method in the next section. Therefore, we can reformulate problem (23,25) in an approximate fashion in the form of a multiple set-point problem for the set-points $\{s_i\}_{i=1}^N$:

$$\min_{y_i, p} \sum_{i=1}^N f(y_i, p, s_i) \omega_i \quad (26)$$

$$\text{s.t. } c(y_i, p, s_i) = 0, \quad \forall i \in \{1, \dots, N\} \quad (27)$$

$$h(y_i, p, s_i) \geq 0, \quad \forall i \in \{1, \dots, N\}. \quad (28)$$

where ω_i denote the quadrature weights. We will investigate this formulation later on.

3.3 Chance Constraint Formulations

Chance constraints leave some flexibility with respect to the inequality restrictions (cf. Ref. [37]). The inequality restrictions are only required to hold with a certain probability \mathcal{P}_0

$$\min_{y, p} \int_{\Omega} f(y, p, s(\zeta)) d\mathcal{P}(\zeta) \quad (29)$$

$$\text{s.t. } c(y, p, s(\zeta)) = 0, \quad \forall \zeta \in \Omega \quad (30)$$

$$\mathcal{P}(\{\zeta \mid h(y, p, s(\zeta)) \geq 0\}) \geq \mathcal{P}_0 \quad (31)$$

So far, chance constraints are used mainly for weakly nonlinear optimization problems Ref. [22, 20]. In the context of structural optimization (which is typically a bilinear problem), this formulation is also called reliability-based design optimization. For more complex problems, we need again some simplification. In Ref. [38] this is performed by applying a Taylor series expansion about a nominal set-point $s^0 := \mathbb{E}(s)$, which is assumed to be equal to the expected value of the random vector s . Suppressing further arguments (y, p) for the moment, the Taylor approximation of 2nd order of f in (29) gives

$$\hat{f}(s) := f(s^0) + \frac{\partial f(s^0)}{\partial s} (s - s^0) + \frac{1}{2} (s - s^0)^T \frac{\partial^2 f(s^0)}{\partial s^2} (s - s^0)$$

Integrating this, we observe

$$\mathbb{E}(f) \doteq \int_{\Omega} \hat{f}(s) d\mathcal{P}(\zeta) = f(s^0) + \frac{1}{2} \sum_{i=1}^k \frac{\partial^2 f(s^0)}{\partial s_i^2} \text{Var}(s_i)$$

where $\text{Var}(s_i)$ is the variance of the i th component of s . Obviously, a first order Taylor series approximation would not give any influence of the stochastic information, which is the reason, why we use an approximation of second order for the objective. In order to deal with the probabilistic chance constraint (31), we also have to approximate its probability distribution. Since the uncertainties are all assumed to be Gaussian and truncated Gaussian, respectively, we use a first order Taylor approximation of the inequality constraint, since we know that this is again Gaussian and truncated Gaussian distributed (unlike the second order approximation) (cp. Ref. [17])

$$\hat{h}(s) := h(s^0) + \frac{\partial h(s^0)}{\partial s}(s - s^0) \sim \frac{1}{\text{const}} N(h(s^0), \sigma_h^2) \cdot \mathbf{1}_{R_h}$$

where we assume for simplicity that h is scalar valued.

Now we can put the Taylor approximations together and achieve a deterministic optimization problem. Since the flow model (30) depends also on the uncertainties s , we should be aware that the derivatives with respect to s mean total derivatives. We express this by reducing the problem in writing $y = y(p, s)$ via (30).

$$\min_p f(y(p, s^0), s^0) + \frac{1}{2} \sum_{i=1}^k \frac{\partial^2 f(y(p, s^0), s^0)}{\partial s_i^2} \text{Var}(s_i) \quad (32)$$

$$\text{s.t. } \mathcal{P}(\{\zeta \mid \hat{h}(y(p, s(\zeta)), s(\zeta)) \geq 0\}) \geq \mathcal{P}_0 \quad (33)$$

For the computation of the total derivatives we can introduce a sensitivity equation as in Ref. [43].

As an example, again we look at the case that s is scalar valued, i.e.

$s \sim \frac{1}{\text{const}_s} N(v, \sigma^2) \cdot \mathbf{1}_{[l, u]}$, where $\text{const}_s = \int_l^u \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-v)^2}{2\sigma^2}\right) dx$ is the scaling factor to normalize the density function. Hence, we obtain the distribution of the probabilistic constraint

$$\hat{h}(s) \sim \frac{1}{\text{const}_{\hat{h}}} N\left(h(s^0), \left(\frac{\partial h(s^0)}{\partial s}\right)^2 \sigma^2\right) \cdot \mathbf{1}_{\left[\frac{\partial h(s^0)}{\partial s} l + h(s^0), \frac{\partial h(s^0)}{\partial s} u + h(s^0)\right]}$$

$$\text{where } \text{const}_{\hat{h}} = \frac{1}{\sqrt{2\pi\left(\frac{\partial h(s^0)}{\partial s}\right)^2 \sigma^2}} \int_{\frac{\partial h(s^0)}{\partial s} l + h(s^0)}^{\frac{\partial h(s^0)}{\partial s} u + h(s^0)} \exp\left(-\frac{(x - h(s^0))^2}{2\left(\frac{\partial h(s^0)}{\partial s}\right)^2 \sigma^2}\right) dx.$$

Finally, the following equivalent representations of the chance constraint

$$\begin{aligned} & \mathcal{P}(\{\zeta \mid \hat{h}(y(p, s(\zeta)), s(\zeta)) \geq 0\}) \geq \mathcal{P}_0 \\ \iff & \mathcal{P}(\{\zeta \mid \hat{h}(y(p, s(\zeta)), s(\zeta)) \leq 0\}) \leq 1 - \mathcal{P}_0 \end{aligned}$$

lead to the deterministic optimization problem

$$\begin{aligned} & \min_p f(y(p, s^0), s^0) + \frac{1}{2} \frac{\partial^2 f(y(p, s^0), s^0)}{\partial s^2} \text{Var}(s) & (34) \\ \text{s.t. } & \frac{1}{\text{const}_{\hat{h}} \sqrt{2\pi \left(\frac{\partial h(s^0)}{\partial s}\right)^2 \sigma^2}} \int_{\frac{\partial h(s^0)}{\partial s} l + h(s^0)}^0 \exp\left(-\frac{(x - h(s^0))^2}{2\left(\frac{\partial h(s^0)}{\partial s}\right)^2 \sigma^2}\right) dx \leq 1 - \mathcal{P}_0 & (35) \end{aligned}$$

The propagation of the input data uncertainties is estimated by the combination of a Second Order Second Moment (SOSM) method and first order Taylor series approximation presented for example in Ref. [38]. Since there is no closed form solution for the integral, the chance constraint is evaluated by a numerical quadrature formula.

Considering geometry uncertainties, a high amount of computational effort arises due to the high dimensionality of the resulting robust optimization problem. In the following, we will introduce two techniques in order to reduce the complexity of the problem: a goal oriented choice of the Karhunen-Loève basis reducing the dimension of the probability space and (adaptive) sparse grid methods in order to efficiently evaluate the high dimensional integrals.

4 Reduction of the Dimension of the Probability Space Using a Goal-Oriented Karhunen-Loève Basis

The evaluation of the objective function in the robust optimization problem (23) requires the computation of the mean, i.e. the computation of the integral of the random field with respect to its probability measure. Applying the introduced Karhunen-Loève-Approximation, the objective function can be written as the following d-dimensional integral

$$\mathbb{E}(f(y, p, \psi(x, \zeta))) = \int_{\Omega} \cdots \int_{\Omega} (f(y, p, \psi(x, Y_1(\zeta), \dots, Y_d(\zeta)))) d\gamma_1(\zeta) \cdots d\gamma_1(\zeta) \tag{36}$$

where $d\gamma_1(\zeta)$ is the one-dimensional Gaussian measure. So, one term more in the truncated Karhunen-Loève expansion to increase the approximation accuracy results in an integral of one-dimension higher. In order to reduce the computational effort, the orthogonal basis functions $\{z_i\}$ will be chosen goal-oriented, i.e. the individual impact of the eigenvectors on the target functional will be taken into account. This method is well established in the model reduction methods of dynamic systems and the adaptive mesh refinement (cf. [1]). The idea is to develop an error indicator for the individual eigenvectors reflecting the influence on the drag. The introduced error analysis of the Karhunen-Loève-Expansion in section 2.2 only gives the approximation error of the random field ψ , but not of the function of interest

$f(y, p, \psi)$. We propose to use sensitivity information to capture the local sensitivities of the drag with respect to the eigenvectors

$$\eta_i := \frac{df}{dz_i} = -\lambda^\top \frac{\partial c}{\partial z_i} + \frac{\partial f}{\partial z_i}, \quad \forall i = 1, \dots, d \quad (37)$$

where λ solves the adjoint equation. The adjoint equation is independent of i , hence it has to be solved only once and the indicator η_i is numerically cheap to evaluate. Now, the reduced basis $\{\hat{z}_i\}$ can be automatically selected, the eigenvector z_i with a large value η_i have to be kept in the reduced basis, whereas a small value indicates that the basis vector can be rejected from the basis.

5 Adaptive Sparse Grid for High-Dimensional Integration

The mean value of the robust optimization problem depending on the current design vector is required in each iteration of the optimization algorithm. Since we cannot solve this integral analytically, we have to approximate it in appropriate, efficient way. Several possibilities can be found in the literature, the most common are: Monte-Carlo simulation, respectively general sampling methods, full tensor grid interpolation and sparse grid interpolation. Their efficiency depends on the dimension d of the probability space Ω_d and on the properties of the integrand $f(y, p, \psi_d)$. Each of these methods provides an approximation \mathbb{E}_N of the mean value $\mathbb{E}(f)$ by evaluating the function $f(y, p, \psi_d)$ in N integration points $\psi_d^1, \dots, \psi_d^N$ and summing the results $f(y_i, p, \psi_d^i)$ multiplied with the weights $\omega_1, \dots, \omega_N$ up

$$\mathbb{E}_N = \sum_{i=1}^N \omega_i \cdot f(y_i, p, \psi_d^i) \quad (38)$$

The sampling methods randomly select realizations of the uncertainties in the given probability space and take some kind of average of the function values at these points which converges to the exact value of the integral due to the law of large numbers. The advantage of this approach consists of the straightforward implementation, the algorithm only needs the underlying integration space as input and function evaluations at the randomly selected points. But on the other hand, the expected convergence rate $O(N^{-\frac{1}{2}})$ requires a large number of function evaluations to ensure a given error tolerance. In our application, one function evaluation is very expensive since the solution of the flow equation, Euler or Navier Stokes equation, is needed. So, the sampling methods, even the improved methods which use additional information in order to select the realizations, are not an appropriate choice in our case to compute the mean value in our optimization problem.

Another possibility obtaining the objective value is the full tensor grid quadrature derived from the full tensor product of the one-dimensional interpolation formulas. Constructing the multi-dimensional interpolation, we first consider the following one-dimensional interpolation formula in order to approximate a function $h : [-1, 1] \rightarrow \mathbb{R}$:

$$Q^i(h) = \sum_{j=1}^N h(Y_j^i) \cdot a_j^i \tag{39}$$

with the set of interpolation points $X^i = \{Y_j^i | Y_j^i \in [-1, 1], j = 1, 2, \dots, m_i\}$, m_i is the number of elements of the set X^i and $a_j^i \equiv a_j(Y_j^i)$ are the interpolation functions.

$$(Q^{i_1} \otimes \dots \otimes Q^{i_d})(f) = \sum_{j_1=1}^{m_1} \dots \sum_{j_d=1}^{m_d} f(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}) \cdot (a_{j_1}^{i_1} \otimes \dots \otimes a_{j_d}^{i_d}) \tag{40}$$

This generalization of the one dimensional formula to the full tensor interpolation (40) provides an approximation of $f : [-1, 1]^d \rightarrow \mathbb{R}$ by evaluating the function f on the regular mesh $X^{i_1} \times \dots \times X^{i_d}$. Considering the difference formulas defined by

$$\Delta^i := Q^{i+1} - Q^i \tag{41}$$

$$Q^0 := 0 \tag{42}$$

(40) can be reformulated as

$$(Q^{i_1} \otimes \dots \otimes Q^{i_d})(f) = \sum_{i_1, \dots, i_d \leq k} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_d})(f) \tag{43}$$

The mean value is then derived from the following equation:

$$\mathbb{E}_N(f) = \sum_{j_1=1}^{m_1} \dots \sum_{j_d=1}^{m_d} f(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}) \cdot \int_{[-1, 1]^d} (a_{j_1}^{i_1} \otimes \dots \otimes a_{j_d}^{i_d})(\mathbf{Y}) d\mathbf{Y} \tag{44}$$

The approximation error for functions with bounded derivatives up to order r has a behaviour of $O(N^{-\frac{r}{d}})$ [8]. Thus, due to the exponential growth of the effort with increasing dimension, this method is not suitable for high stochastic dimensions, which is the case in our application. To circumvent this curse of dimensionality, we apply a sparse grid method in order to preserve the accuracy of the tensor grid quadrature, but avoiding the exponential growth of interpolation nodes.

The underlying idea of sparse grids was originally found by the Russian mathematician Smolyak [40]. The sparse interpolant is given as [33]:

$$S(k, d)(f) = \sum_{k-d+1 \leq |\mathbf{i}| \leq k} (-1)^{k-|\mathbf{i}|} \cdot \binom{d-1}{k-|\mathbf{i}|} \cdot (Q^{i_1} \otimes \dots \otimes Q^{i_d})(f) \tag{45}$$

with $k \geq d$, $\mathbf{i} \in \mathbb{N}^d$ multi-index and $|\mathbf{i}| = \sum_{j=1}^d i_j$. The index i_j indicates the order in the j th dimension, so the algorithm combines only those one-dimensional quadrature formulas whose indices fulfill the constraint that the total sum across all dimensions is greater or equal than $k - 1 + d$ and smaller or equal than k . Using incremental interpolation formulas Δ^i , (45) can be transformed to

$$\begin{aligned}
 S(k, d)(f) &= \sum_{|\mathbf{i}| \leq k} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_d})(f) \\
 &= S(k-1, d)(f) + \sum_{|\mathbf{i}|=k} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_d})(f)
 \end{aligned}$$

with $\Delta^i = Q^{i+1} - Q^i$, $Q^0 \equiv 0$ and $S(d-1, d) \equiv 0$. The collection of all the interpolation points

$$\mathcal{H}(k, d) = \bigcup_{k-d+1 \leq |\mathbf{i}| \leq k} (X^{i_1} \times \dots \times X^{i_d}) \tag{46}$$

is called a sparse grid of level k .

The derivation of the sparse grid suggests the use of nested interpolation functions due to the recursive construction. In the literature, the most popular choice of the collocation points is the Clenshaw-Curtis grid at the non-equidistant extrema of the Chebyshev polynomials and the underlying interpolation formula is the Chebyshev-Gauss-Lobatto formula.

5.1 Adaptive Sparse Grid

Since the function evaluations are very expensive in our application, we introduce in this section an adaptive sparse grid strategy in order to further reduce the number of grid points but conserving the approximation quality. The presented isotropic Smolyak algorithm is effective for problems whose input data uniformly depend on all dimensions. But the convergence rate deteriorates for highly anisotropic problems, such as those appearing when the input random variables come from a Karhunen-Loève-Expansion as in our application [9]. The reduction of computational effort can be achieved by using spatially adaptive or dimension-adaptive refinement [33], [14]. In order to develop adaptive schemes during the cubature process, the interpolation error can be used as an adaptivity indicator. Therefore, nested cubature formulas are useful since they allow the error evaluation based on the difference of two subsequent formulas. Due to the fact that in our application the mean value is computed by the sparse grid interpolation, this target value is also used as an error indicator for the adaptivity. The dimension-adaptive quadrature method tries to find important dimensions and adaptively refines in this with respect to given error estimators. This leads to an approach which is based on generalized sparse grid index sets [14]. This strategy allows to employ every nested interpolation formulas, so it can be chosen problem dependent, e.g. in our application depending on the distribution of the random variables. On the other hand, the locally refined sparse grid gives more flexibility in the adaptive procedure, but requires equidistant support nodes. In the following, we will discuss both strategies and compare the resulting sparse grids in the numerical results later on.

5.1.1 Locally Refined Adaptive Sparse Grid

Below, we introduce a locally adaptive hierarchical sparse grid approach using piecewise multilinear hierarchical basis functions following closely [27], [33]. Due to the straightforward implementation of the refinement, we choose the linear hat functions as interpolation basis functions which are also well established in the adaptive mesh refinement. The support nodes of the one-dimensional basis function are given by

$$Y_j^i = \begin{cases} 0 & \text{for } j = 1, m_i = 1 \\ 2 \cdot \frac{j-1}{m_i-1} - 1 & \text{for } j = 1, \dots, m_i, m_i > 1 \end{cases} \quad (47)$$

with

$$m_i = \begin{cases} 1, & \text{for } i = 1 \\ 2^{i-1} + 1 & \text{for } i > 1 \end{cases} \quad (48)$$

Hence, the interpolation formulas are defined by

$$a_j^i(Y) = \begin{cases} 1 - \frac{1}{2}(m_i - 1) \cdot |Y - Y_j^i|, & \text{if } |Y - Y_j^i| < \frac{2}{m_i-1} \\ 0, & \text{otherwise.} \end{cases} \quad (49)$$

The discussed univariate nodal basis functions (49) are now transformed into multivariate hierarchical basis functions which are fundamental for the adaptive sparse grid. Considering once again the difference formula

$$\Delta^i(h) = Q^{i+1}(h) - Q^i(h) \quad (50)$$

with

$$Q^i(h) = \sum_{Y_j^i \in G^i} a_j^i \cdot h(Y_j^i) \quad (51)$$

we obtain due to the fact that the support nodes are nested (e.g. $G^i \subset G^{i+1}$) and accordingly $Q^{i-1}(h) = Q^i(Q^{i-1}(h))$ the following representation of (50)

$$\Delta^i(h) = \sum_{Y_j^i \in G^i} a_j^i \cdot h(Y_j^i) - \sum_{Y_j^i \in G^i} a_j^i \cdot Q^{i-1}(h)(Y_j^i) \quad (52)$$

$$= \sum_{Y_j^i \in G^i} a_j^i \cdot (h(Y_j^i) - Q^{i-1}(h)(Y_j^i)) \quad (53)$$

$$= \sum_{Y_j^i \in G_\Delta^i} a_j^i \cdot (h(Y_j^i) - Q^{i-1}(h)(Y_j^i)) \quad (54)$$

since $h(Y_j^i) - Q^{i-1}(h)(Y_j^i) = 0, \quad \forall Y_j^i \in G^{i-1}$. Renumbering the elements in $G_\Delta^i = G^i \setminus G^{i-1}$, with $m_\Delta^i = \#G_\Delta^i = m_i - m_{i-1}$, leads to

$$\Delta^i(h) = \sum_{j=1}^{m_i^{\Delta}} a_j^i \cdot (h(Y_j^i) - Q^{i-1}(h)(Y_j^i)). \quad (55)$$

We define $w_j^i = h(Y_j^i) - Q^{i-1}(h)(Y_j^i)$ as the 1D hierarchical surplus [6] which is the difference between the current and previous interpolation level. If the 1D Grid of level k interpolates the function h exactly, w_j^k is equal to zero for all j . So, this value can be used as an error indicator for each inserted grid point, since the hierarchical surpluses tend to zero as the level k tends to infinity (for continuous functions). Considering now again the multi-dimensional case, we obtain the sparse grid [46] in hierarchical form applying the derived formula of Δ^i .

$$S(k, d)(f) = S(k-1, d)(f) + \sum_{|\mathbf{i}|=k} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_d})(f) \quad (56)$$

$$= S(k-1, d)(f) + \Delta S(k, d)(f) \quad (57)$$

with

$$\begin{aligned} \Delta S(k, d)(f) &= \sum_{|\mathbf{i}|=k} \sum_{\mathbf{j} \in B_{\mathbf{i}}} \underbrace{(a_{j_1}^{i_1} \otimes \dots \otimes a_{j_d}^{i_d})}_{a_{\mathbf{j}}^{\mathbf{i}}} \cdot \\ &\quad \cdot \underbrace{(f(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}) - S(k-1, d)(f)(Y_{j_1}^{i_1}, \dots, Y_{j_d}^{i_d}))}_{w_{\mathbf{j}}^{\mathbf{i}}} \end{aligned} \quad (58)$$

where $B_{\mathbf{i}} := \{\mathbf{j} \in \mathbb{N}^{\mathbb{N}} : Y_{j_l}^{i_l} \in G_{\Delta}^{i_l} \text{ for } j_l = 1, \dots, m_{\Delta}^{i_l}, l = 1, \dots, d\}$ is a new set of multi-indices consistent with the multivariate hierarchical basis $\{a_{\mathbf{j}}^{\mathbf{i}} : \mathbf{j} \in B_{\mathbf{i}}, \mathbf{1} \leq \mathbf{i}\}$.

Thus, the objective function in our application can be approximated by the following rather abstract expression:

$$f(p, \psi_d(\zeta)) = \sum_{|\mathbf{i}| \leq k} \sum_{\mathbf{j} \in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}(p) \cdot a_{\mathbf{j}}^{\mathbf{i}}(\zeta) \quad (59)$$

The mean value of the objective function can be then computed as:

$$\mathbb{E}_N(f(p)) = \sum_{|\mathbf{i}| \leq k} \sum_{\mathbf{j} \in B_{\mathbf{i}}} w_{\mathbf{j}}^{\mathbf{i}}(p) \cdot \int_{\Omega} a_{\mathbf{j}}^{\mathbf{i}}(\psi_d(\zeta)) d\mathcal{P}(\zeta) \quad (60)$$

where

$$\int_{-1}^1 a_j^i(Y) dY = \begin{cases} 1 \cdot l, & \text{if } i = 1, \\ \frac{1}{2} \cdot l, & \text{if } i = 2, \\ 2^{1-i} \cdot l, & \text{otherwise.} \end{cases} \quad (61)$$

where l denotes the length of the given 1D interval, that means in our example $l = 2$. Instead of using the hierarchical surplus $w_{\mathbf{j}}^{\mathbf{i}}$ as an error indicator for the adaptivity (cf. [27], [33]), we suggest to adapt the grid checking the following expression:

$$\tilde{w}_j^i := w_j^i \cdot \int_{\Omega} a_j^i(\psi_d(\zeta)) d\mathcal{P}(\zeta) \quad (62)$$

Since it is not necessary to exactly interpolate the drag depending on the uncertainty in the optimization loop in our application, the introduced adaptivity indicator \tilde{w}_j^i only measures the difference between the value of the mean inserting a new point Y_j^i of the current level of interpolation and the corresponding value of the mean at the previous interpolation level. The resulting algorithm in order to construct the sparse grid which is then used for the optimization slightly differs from [27], [33] due to the modification in the adaptivity indicator.

5.1.2 Dimension Adaptive Sparse Grid

The main advantage of the dimension adaptive refinement strategy is the fact that one can use problem dependent quadrature formulas in order to construct the adaptive sparse grid. In our application, the objective function, the drag, is multiplied by the Gaussian density function, so that Gaussian Hermite polynomials are optimal with respect to the weighting function.

First, a generalization of sparse grids will be introduced which allows to weight the dimensions according to their importance on the target functional. The idea of generalized sparse grids and especially dimension adaptive sparse grid can be found in [14], [12], [7] and [26]. The original sparse grid of order k combines all the incremental functions which sum up to order k , that means only those indices are considered which are contained in the unit simplex $|\mathbf{i}| \leq k$. [14] and [13] suggest to allow a more general index set which can be then adaptively chosen with respect to the importance of each dimension.

An index set I is called *admissible* if $\forall \mathbf{i} \in I$

$$\mathbf{i} - e_j \in I, \quad \forall 1 \leq j \leq d, k_j > 1,$$

where $e_j \in \mathbb{R}^d$ is the j th unit vector. The generalized index set I contains for an index \mathbf{i} all indices which have smaller entries in one or more dimensions. Due to this fact, the incremental sparse grid formula (46) is still well defined for the new index sets and is given as

$$S(k, d)(f) = \sum_{\mathbf{i} \in I} (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_d})(f). \quad (63)$$

The generalized definition of sparse grids includes the original sparse grid and the full tensor grid definition (cf. (46), (43)). Further, equation (63) particularly leaves more flexibility to the choice of the grids and therefore allows to handle anisotropic problems which emphasize the following example of an admissible index set in \mathbb{R}^2 :

$$I = \left\{ \binom{1}{1}, \binom{2}{1}, \binom{3}{1}, \binom{4}{1}, \binom{5}{1}, \binom{1}{2} \right\}.$$

This example of an admissible index set i shows the feasibility of an refinement in only one dimension (here in the first dimension) which is the required feature for the adaptivity.

If a priori knowledge of the underlying function is available, an admissible index set with respect to the additional information can be chosen. Since this is not the case in our application, an algorithm is introduced in the following which automatically computes an admissible index set in a dimension adaptive way (cf. [14], [13]). Therefor, we start with the coarsest sparse grid, that means $I = \{(0, \dots, 0)\}$ and successively add new indices such that

- the new index set remains admissible
- the approximation error is reduced.

For the second point, an error indicator is needed. Taking a look at the difference formula (63), the term

$$\Delta_{\mathbf{i}} = (\Delta^{i_1} \otimes \dots \otimes \Delta^{i_d})(f) \quad (64)$$

indicates the reduction in the approximated integral for each new added index. [14] suggests to further involve the number of function evaluations to avoid an too early stopping. Since equation (64) shows good results in our application, we directly use $\Delta_{\mathbf{i}}$ as an error indicator for the adaptivity.

As mentioned before, the main advantage of the dimension adaptive refinement is the fact that the quadrature formulas can be chosen problem dependent. Considering geometry uncertainties in the robust optimization, the Karhunen-Loève expansion leads to the following objective function

$$\mathbb{E}(f(y, p, \psi(x, \zeta))) \doteq \int_{\Omega} \dots \int_{\Omega} (f(y, p, \psi_d(x, Y_1(\zeta), \dots, Y_d(\zeta))) d\varphi_1(\zeta) \dots d\varphi_1(\zeta),$$

so that the Gauß-Hermite formulas are an appropriate choice for the quadrature. The one dimensional Hermite polynomials are orthogonal polynomials over $(-\infty, \infty)$ with the weight function $\omega(x) = \exp(-x^2)$. The Gauß-Hermite quadrature belongs to the class of Gaußformulas which are constructed by choosing both the points and the weights with the goal to exactly integrate as many polynomials as possible. The Gaußformulas achieve the polynomial exactness of $2n - 1$ where n is the number of abscissas of the quadrature rule. In GaußHermite quadrature the integral of the form $\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx$ is approximated by

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2) dx \approx \sum_{i=1}^m \omega_i f(x_i) \quad (65)$$

where the abscissas x_i are zeros of the m th Hermite polynomial and the ω_i are the corresponding weights. The Hermite polynomials are defined as

$$H_n(x) = (-1)^n \exp(x^2) \frac{d^n}{dx^n} \exp(-x^2) \quad (66)$$

and the weights

$$\omega_i = \frac{2^{n-1} n! \sqrt{\pi}}{n^2 H_{n-1}(x_i)^2}. \quad (67)$$

The Gauß-Hermite quadrature formulas are weakly nested, that means the rules of odd order all include the abscissa 0. Since the nesting is a favorable feature constructing the sparse grid, this property will be taken into account. For the numerical results, we will use the Gauß-Hermite quadrature formulas of order 1, 3, 7, 15 to construct the dimension adaptive sparse grid.

6 One-Shot Aerodynamic Shape Optimization and Its Coupling to Robust Design

Novel one-shot aerodynamic shape optimization in the form (14-16) have been introduced in [19, 18]. They have the potential of fast convergence in only a small multiple of cpu-time compared to on flow simulation. These methods are based on approximate reduced SQP iterations in order to generate a stationary point satisfying the first order KKT optimality conditions.

In this context, a full SQP-approach reads as

$$\begin{bmatrix} \mathcal{L}_{yy} & \mathcal{L}_{yp} & h_x^\top & c_x^\top \\ \mathcal{L}_{py} & \mathcal{L}_{pp} & h_p^\top & c_p^\top \\ h_x & h_p & 0 & 0 \\ c_x & c_p & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_y^\top \\ -\mathcal{L}_p^\top \\ -h \\ -c \end{pmatrix}, \quad \begin{pmatrix} y^{k+1} \\ p^{k+1} \\ \mu^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ p^k \\ \mu^k \\ \lambda^k \end{pmatrix} + \tau \cdot \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} \quad (68)$$

The symbol \mathcal{L} denotes the Lagrangian function. We assume that the lift constraint h is active at the solution, which is the reason that we formulate is rather as an equality condition in the single setpoint case. The approach (68) is not implementable in general, because one usually starts out with a flow solver for $c(y, p) = 0$ and seeks a modular coupling with an optimization approach, which does not necessitate to change the whole code structure, as would be the case with formulation (68). A modular but nevertheless efficient alternative is an approximate reduced SQP approach as justified in [16].

$$\begin{bmatrix} 0 & 0 & 0 & A^\top \\ 0 & B & \gamma & c_p^\top \\ 0 & \gamma^\top & 0 & 0 \\ A & c_p & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_y^\top \\ -\mathcal{L}_p^\top \\ -h \\ -c \end{pmatrix}, \quad \begin{pmatrix} y^{k+1} \\ p^{k+1} \\ \mu^{k+1} \\ \lambda^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ p^k \\ \mu^k \\ \lambda^k \end{pmatrix} + \tau \cdot \begin{pmatrix} \Delta y \\ \Delta p \\ \Delta \mu \\ \Delta \lambda \end{pmatrix} \quad (69)$$

where

$$\gamma = h_p^\top + c_p^\top \alpha, \text{ such that } A^\top \alpha = -h_x^\top$$

The matrix A denotes an appropriate approximation of the system matrix c_x , which is used in the iterative forward solver. An algorithmic version of this modular formulation is given by the following steps

- (1) generate λ^k by performing N iterations of an adjoint solver with right hand side $f_y^\top(y^k, p^k)$ starting in λ^k
- (2) generate α^k by performing N iterations of an adjoint solver with right hand side $h_y^\top(y^k, p^k)$ starting in α^k
- (3) compute approximate reduced gradients

$$g = f_p^\top + c_p^\top \lambda^{k+1}, \quad \gamma = h_p^\top + c_p^\top \alpha^{k+1}$$

- (4) generate B_{k+1} as an approximation of the (consistent) reduced Hessian
- (5) solve the QP

$$\begin{bmatrix} B & \gamma \\ \gamma^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} -g \\ -h \end{pmatrix}$$

- (6) update $p^{k+1} = p^k + \Delta p$
- (7) compute the corresponding geometry and adjust the computational mesh
- (8) generate y^{k+1} by performing N iterations of the forward state solver starting from an interpolation of y^k at the new mesh.

This highly modular algorithmic approach is not an exact transcription of equation (69), but is shown in [16] to be asymptotically equivalent and to converge to the same solution. The overall algorithmic effort for this algorithm is typically in the range of factor 7 to 10 compared to a forward stationary simulation.

Now we generalize this algorithmic framework to the semi-infinite problem formulation (23)-(25). Numerical approaches to this problem class have already been proposed in [2, 3, 4].

For the sake of simplicity, we restricted the formulation to a problem with two set-points coupled via the objective, which is a weighted sum of all set-point objectives (weights: ω_1, ω_2), and via the free optimization variables p , which are the same for all set-points. The generalization to more setpoints (i.e., the adaptive sparse grid points below) is then obvious. The lift constraint is formulated for the smallest value s^{\min} of all grid points. The corresponding Lagrangian in our example is

$$\mathcal{L}(y_1, y_2, p, \lambda_1, \lambda_2) = \sum_{i=1}^2 \omega_i f_i(y_i, p, s^i) + \sum_{i=1}^2 \lambda_i^\top c_i(y_i, p, s^i) + \mu h(y_{\min}, p, s^{\min}) \quad (70)$$

The approximate reduced SQP method above applied to this case can be written in the following form

$$\begin{bmatrix} 0 & 0 & 0 & 0 & A_1^\top & 0 \\ 0 & 0 & 0 & 0 & 0 & A_2^\top \\ 0 & 0 & B & \gamma & c_{1,p}^\top & c_{2,p}^\top \\ 0 & 0 & \gamma_1 & 0 & 0 & 0 \\ A_1 & 0 & c_{1,p} & 0 & 0 & 0 \\ 0 & A_2 & c_{2,p} & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} \Delta y_1 \\ \Delta y_2 \\ \Delta p \\ \Delta \mu \\ \Delta \lambda_1 \\ \Delta \lambda_2 \end{pmatrix} = \begin{pmatrix} -\mathcal{L}_{y_1}^\top \\ -\mathcal{L}_{y_2}^\top \\ -\mathcal{L}_p^\top \\ -h \\ -c_1 \\ -c_2 \end{pmatrix} \quad (71)$$

We notice that the linear sub-problems involving matrices A_i^\top are to be solved independently, and therefore trivially in parallel. The information from all these parallel adjoint problems is collected in the reduced gradient

$$g = \sum_{i=1}^2 \omega_i f_p^\top + \sum_{i=1}^2 c_p^\top \lambda_i$$

Next, the solution of optimization step

$$\begin{bmatrix} B & \gamma_1 \\ \gamma_1^\top & 0 \end{bmatrix} \begin{pmatrix} \Delta p \\ \mu^{k+1} \end{pmatrix} = \begin{pmatrix} -g \\ -h \end{pmatrix}$$

is distributed to all approximate linearized forward problems

$$A_i \Delta y_i + c_{i,p} \Delta p = -c_i,$$

which can then again be performed in parallel.

7 Numerical Results

The numerical results include a numerical comparison of the introduced robust formulations considering the optimization of a transonic RAE2822 profile under scalar-valued uncertainties. Further, the optimal aerodynamic shape under geometrical uncertainties is computed in a 2d Euler and Navier-Stokes testcase comparing the adaptive refinement strategies of sparse grids. The last section shows the influence of shape uncertainties on the quantities of interest in the 3d testcase SFB-401.

7.1 Numerical Comparison of the Introduced Robust Formulations

We investigate the shape optimization of a RAE2822 profile in transonic Euler flow, by the use of the CFD software FLOWer provided by DLR within a one-shot framework. The block-structured FLOWer code solves the three-dimensional compressible Reynolds-averaged Navier-Stokes equation in integral form and provides different turbulence models. The equations are solved by a finite-volume method with second order upwind or central space discretization. The discrete equations are integrated explicitly by multistage Runge-Kutta schemes, using local time stepping and multigrid acceleration. In our example, the space is discretized by a 133×33 grid, see Fig. [1](#). For parametrization, the airfoil is decomposed into thickness and camber distribution. Then, only the camber of the airfoil is parametrized by 21 Hicks-Henne functions and the thickness is not changed during the optimization process.

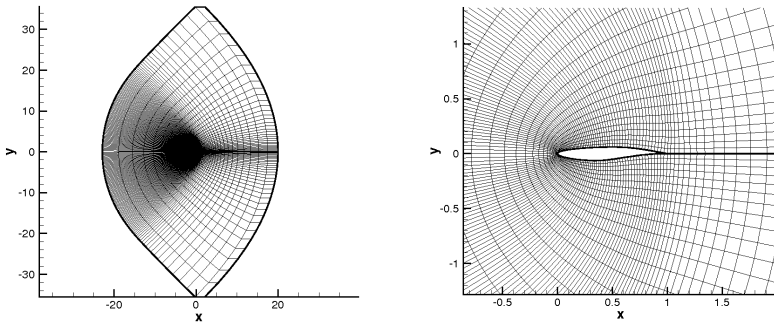


Fig. 1 Grid for the RAE2822 airfoil: the total geometrical plane and zoom around the airfoil

In this section, we consider the Mach number as an uncertain parameter. The Mach number is assumed to be in the range of $[0.7, 0.76]$. Under the assumption of truncated normally distributed parameter, we obtain the density function shown below in the Fig. 2

At first, we perform numerical comparisons between a single set-point problem formulation at the setpoint $s^0 = 0.73Mach$ with the robust formulations in sections 3.2 and 3.3. In particular, we compare four formulations: (1) non-robust optimization at the Mach number 0.73 (fixed Mach number 0.73), (2) semi-infinite formulation of equations (26-28), (3) chance constraint formulation of equations (34-35) and (4) non-robust optimization at the Mach number 0.73 (fixed Mach number 0.73) but maintaining feasibility over the whole range of perturbations.

The following figures show evaluations of the objective (drag), Fig. 3, in these cases as well as the constraint (lift), Fig. 4

We state the following observations: The semi-infinite robust formulation has a better lift to drag ratio than the chance constraint formulation, in particular in

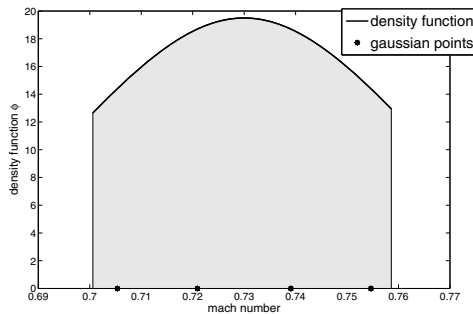


Fig. 2 Density function of the random variables $Ma \sim \frac{1}{const} N(0.73, 0.001) \cdot \mathbf{1}_{[0.7, 0.76]}$ and the corresponding Gaussian points

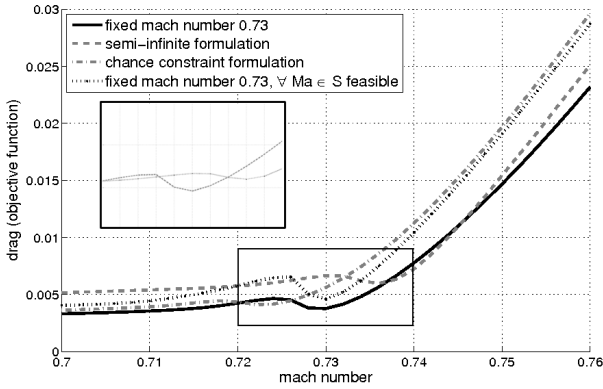


Fig. 3 Comparison of drag

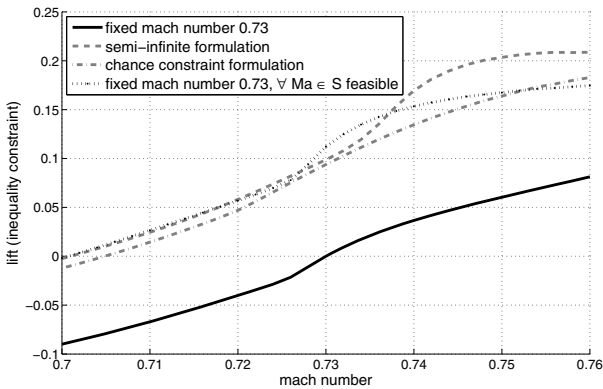


Fig. 4 Comparison of lift constraint

the region above the set-point 0.73, due to the fact that the semi-infinite formulation shows a higher lift over the whole range of variations (cf. Fig. 4) and for the greater part a better drag performance than the chance constraint (cf. Fig. 3). Comparing the robust formulations and the single-set point solution which is feasible over the whole range of variations, the robust solutions leads to a higher drag at the nominal point, but we can observe that the semi-infinite formulation will lead to the best distribution if the Mach number deviates from the set-point. The resulting shapes are shown in Fig. 5. Summing the results up, it can be stated that the semi-infinite formulations leads to a robust solution which gives a little bit higher drag at the nominal point but a better performance over the whole range of variations. Since the semi-infinite formulation seems to be favorable in our application, the following numerical results considering geometrical uncertainties are based on this formulation.

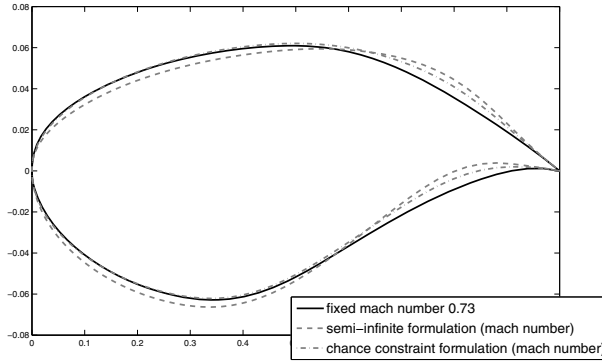


Fig. 5 Resulting shapes of the robust optimization

7.2 Numerical Results Considering Geometrical Uncertainties (Testcase RAE2822)

In this section, we present the numerical results of robust optimization under shape uncertainties of a RAE2822 profile in Euler and Navier-Stokes flow. Since the robust optimization problem is solved within a one-shot framework, we use the flow solver TAU provided by DLR which allows the computation of gradients by the adjoint approach in the Euler and Navier-Stokes case. The TAU Code is a CFD software package for the prediction of viscous and inviscid flows about complex geometries from the low subsonic to the hypersonic flow regime employing hybrid unstructured grids. In the Euler case, the profile is described by 129 surface grid points and in the Navier-Stokes case, by 192 surface grid points. Again, the airfoil is parametrized by 21 Hicks-Henne functions. The geometry uncertainties are characterized by a Gaussian random field and the following second order statistics

$$\mathbb{E}(\psi(x, \zeta)) = 0 \quad \forall x \in \Gamma \tag{72}$$

$$Cov(x, y) = (0.005)^2 \cdot \exp\left(-\frac{\|x-y\|^2}{(0.1)^2}\right) \quad \forall x, y \in \Gamma. \tag{73}$$

One realization of the random field and the resulting perturbed geometry is shown in Fig. 6 and Fig. 7. Representing the random field for the numerical treatment ψ in a finite number of independent random variables using the Karhunen-Loève-Expansion, one has to solve the eigenvalue problem (9). In our two-dimensional testcase, the discretization of the profile leads to a matrix of size (129×129) and (192×192) in the Navier-Stokes case, so the eigenvalues and eigenvectors can be computed by common methods. The distribution of the eigenvalues of the given random field (72-73) is shown in the next Fig. 8. As stated before, the eigenvalues exponentially converge towards 0. For the first numerical results, we have considered only the first four eigenvalues and eigenvectors to represent the random field ψ of

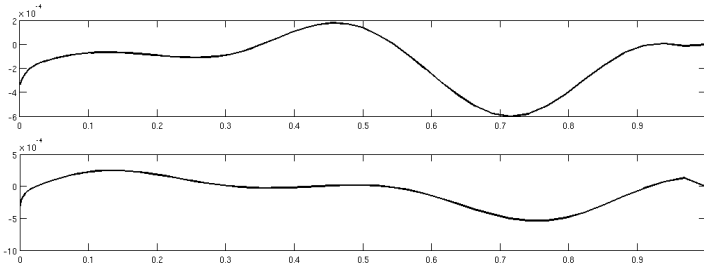


Fig. 6 One realization of the random field ψ : perturbations on the upper side of the profile (above) and on the lower side (below)

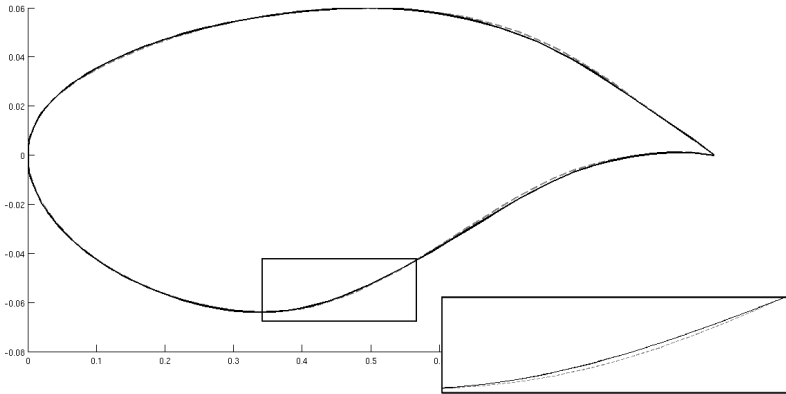


Fig. 7 Resulting perturbed geometry compared with the original shape (dashed line)

perturbations. The corresponding eigenvectors are shown in Fig. 9. Using the truncated Karhunen-Loève representation, the mean value of the drag is then computed by

$$\mathbb{E}(f(y, p, \psi_4(x, \omega))) = \int_{\Omega} \int_{\Omega} \int_{\Omega} \int_{\Omega} (f(y, p, \sum_{i=1}^4 \sqrt{\lambda_i} z_i(x) Y_i(\zeta))) d\gamma_1(\zeta) \cdots d\gamma_1(\zeta) \tag{74}$$

The random variables Y_i are uncorrelated and therefore independent, so one has to approximate a four-dimensional integral in the optimization problem. To further reduce the computational effort, we investigate the influence of the individual eigenvectors in order to reject those eigenvectors from the reduced basis which have no impact on the target functional. Since the following results are problem dependent, we will now distinguish the Euler and Navier-Stokes case.

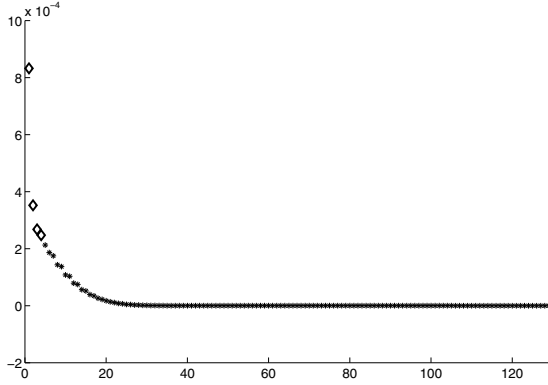


Fig. 8 Distribution of the eigenvalues of the given random field ψ

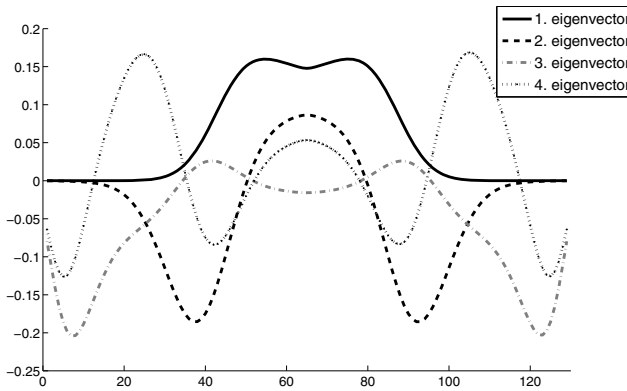


Fig. 9 First four eigenvectors of the given random field ψ

7.2.1 Euler Flow

As Fig. 10 shows, the third eigenvector has no impact on the objective function, hence it can be rejected from the Karhunen-Loève basis and the dimension of the integral is reduced. This behaviour is also reflected by the introduced indicator. Consequently, the mean value is given by

$$\begin{aligned}
 \mathbb{E} \left(f(y, p, \psi_4^{reduced}(x, \zeta)) \right) &= \\
 &= \int_{\Omega} \int_{\Omega} \int_{\Omega} \left(f(y, p, \sum_{\substack{i=1 \\ i \neq 3}}^4 \sqrt{\lambda_i} z_i(x) Y_i(\zeta)) \right) d\gamma_1(\zeta) d\gamma_1(\zeta) d\gamma_1(\zeta)
 \end{aligned} \tag{75}$$

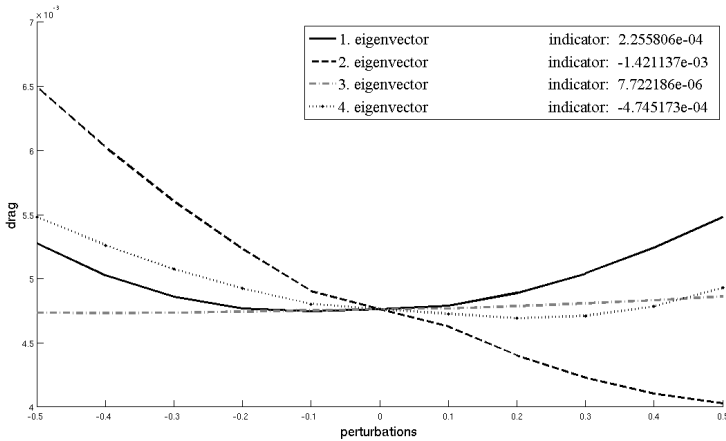


Fig. 10 Drag performance of the first four eigenvectors on the target functional

If one approximate the expected value (175) using a full tensor grid interpolation (40), 729 grid points will be needed to reach the error tolerance of $3 \cdot 10^{-4}$. The resulting full grid is shown in Fig. 11. Since we want to compare the efficiency of the different introduced methods, we have chosen multilinear hierarchical basis function as ansatzfunctions for the sparse and full tensor grid. The sparse grid method can reduce the computational effort by a factor of 10 maintaining the same approximation quality. The corresponding grid is depicted in Fig. 12. As Fig. 12

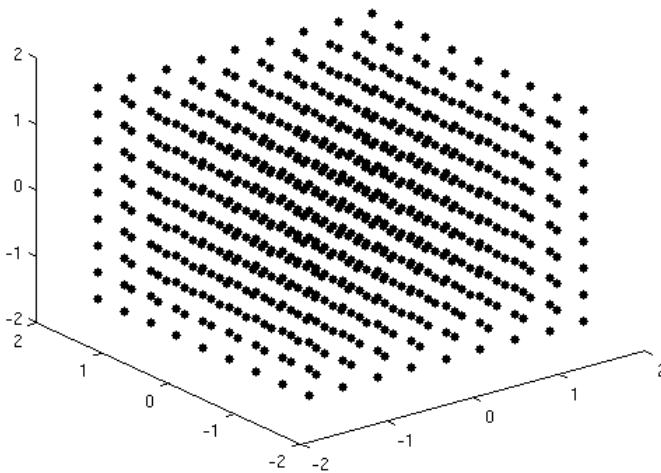


Fig. 11 Full tensor grid with 729 grid points

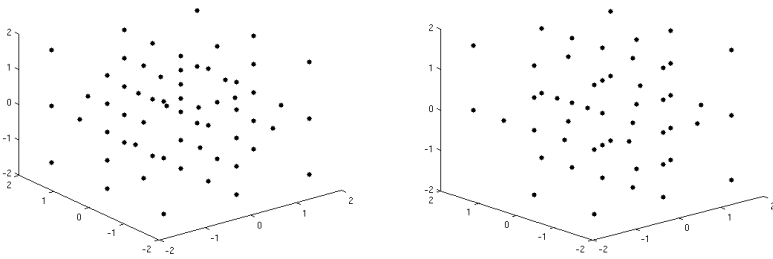


Fig. 12 Sparse grid with 69 grid points and locally refined sparse grid with 52 grid points

shows, the number of grid points can again be reduced from 69 grid points to 52 grid points, i.e. a reduction of 17 flow simulations in each iteration is reached using a local refinement strategy. Since the optimization requires the evaluation of the mean value in each iteration, this reduction by factor 15 compared with the full grid takes place in each step of the optimization algorithm and hence significantly speed up the whole algorithm. The construction of the adaptive sparse grid although needs some additional function evaluations in order to compute the adaptivity indicator, but this amount of computational effort occurs outside the optimization loop, i.e. these costs are negligible.

Beside the local refinement strategy, a dimension adaptive sparse grid was introduced in section 5.1. The main advantage of this approach lies in the possibility to choose the underlying quadrature formulas problem dependent. As discussed before, Gauß-Hermite formulas are used. Due to the higher accuracy of Gauß-Hermite

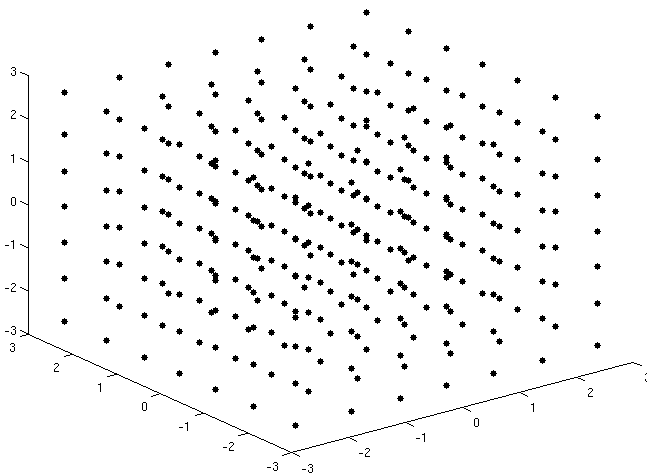


Fig. 13 Full tensor grid with 343 grid points

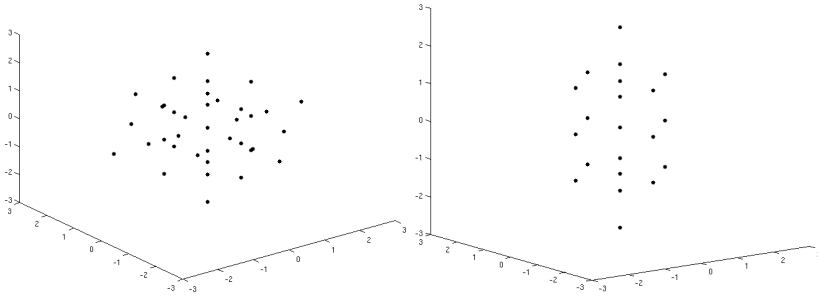


Fig. 14 Sparse grid with 37 grid points and dimension adaptive sparse grid with 21 grid points

quadrature formulas, an error tolerance of 10^{-5} is required. An usual sparse grid can reduce the number of grid points from 343 to 37, again almost a factor of 10. The dimension adaptive strategy results in a grid with 21 points (cf. Fig. 14). The comparison of the two refinement strategies shows that the use of problem dependent quadrature formulas can significantly reduce the size of the grid and increase the approximation quality at the same time. Hence, the objective function in the semi-infinite formulation is approximated by the dimension adaptive grid with 21 grid points. The next two Fig. 15, 16 compare the results of the robust optimization and of the single setpoint optimization, i.e. without considering any uncertainties in the optimization.

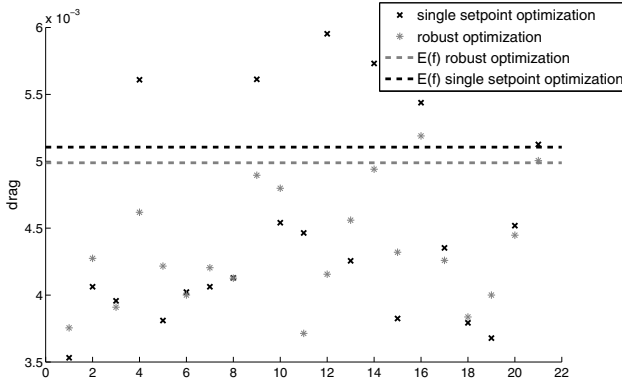


Fig. 15 Drag performance of the 21 perturbed geometries

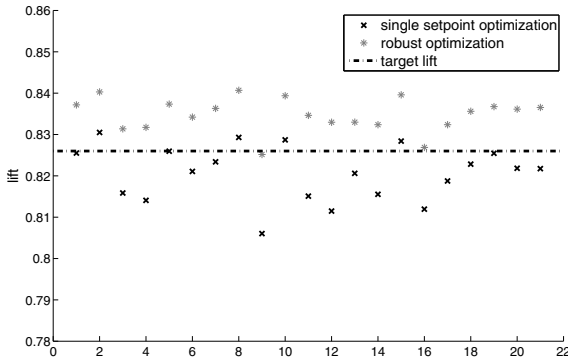


Fig. 16 Lift performance of the 21 perturbed geometries

The drag and lift performance is plotted against the 21 perturbed geometries and the dashed line in Fig. 15 indicates the mean value of the drag. The robust optimization improves the mean value of the target functional and also leads at the same time to a better lift performance over the whole range of perturbations, whereas the single setpoint optimization is infeasible in more than the half of the considered grid points. Summing it all up, it can be said that the robust optimization leads to a better lift to drag ratio than the single setpoint optimization and the resulting profile is more robust against small perturbations of the shape itself. Last, we will compare the different resulting shape in Fig. 17.

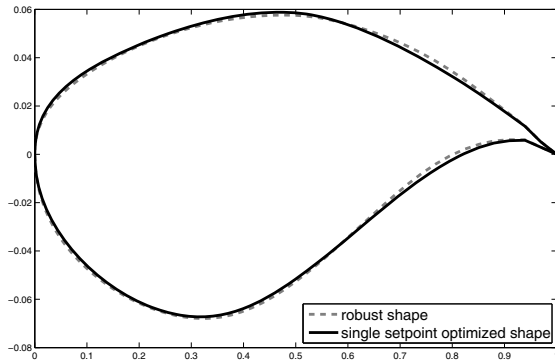


Fig. 17 Comparison of the optimized shapes

Although we have assumed only small perturbations of the shape itself (cf. Fig. 7), the difference between the robust shape and the single setpoint is well recognizable.

7.2.2 Navier-Stokes Flow

As in the Euler case, we investigate the influence of the individual eigenvector of the Karhunen-Loève expansion on the target functional depicted in Fig. 18. Since the third eigenvector has no impact on the target functional, the mean value is approximated using the first, second and fourth basis vector (cf. (75)). If the same error tolerance $3 \cdot 10^{-4}$ as in the previous testcase is required, a full grid of 4913 grid points based on multilinear ansatzfunctions has to be used in order to compute the objective function. A reduction factor of 28 can be achieved by a sparse grid approach which is further improved by a local refinement strategy. The resulting adaptive grid consists of 99 grid points.

Since the computational effort in the Navier-Stokes case is much higher than in the Euler testcase, the number of grid points need to be further reduced in order to make a robust optimization possible. The use of Gauß-Hermite quadrature formulas results in the same full grid and sparse grid shown in Fig. 13 and Fig. 14 as in the Euler testcase fullfilling the error tolerance $3 \cdot 10^{-4}$. The dimension adaptive approach leads to the following grid 21

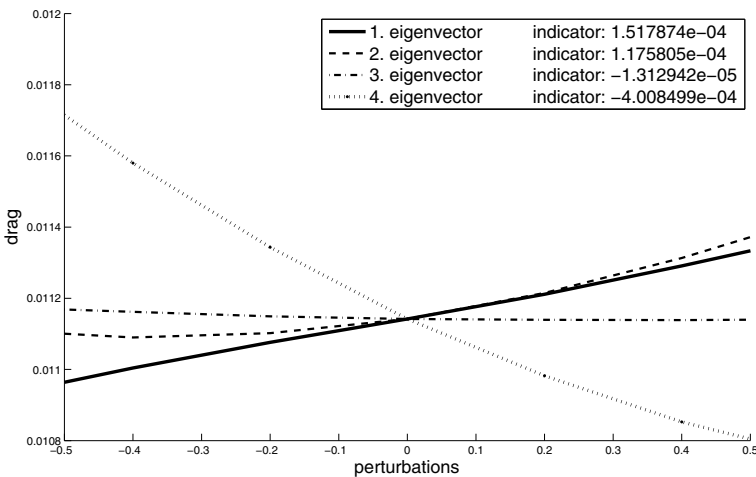


Fig. 18 Drag performance of the first four eigenvectors on the target functional

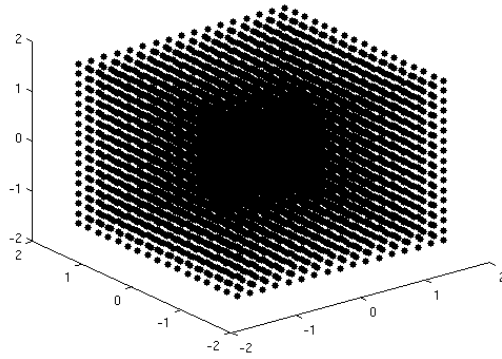


Fig. 19 Full tensor grid with 4913 grid points

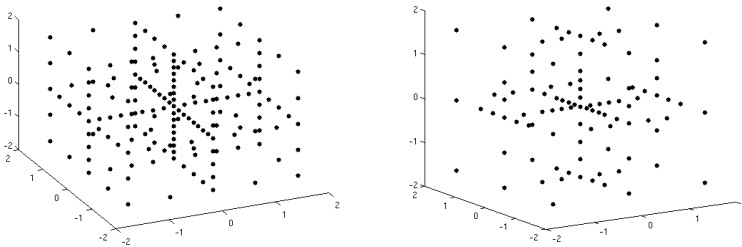


Fig. 20 Sparse grid with 177 grid points and locally refined sparse grid with 99 grid points

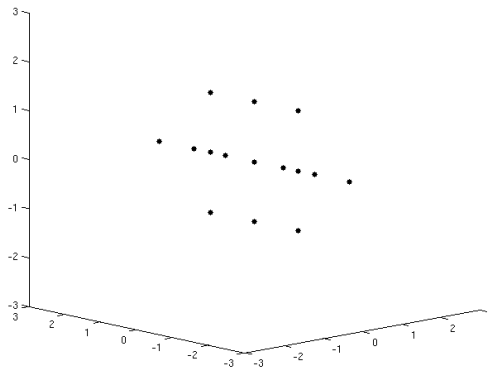


Fig. 21 Dimension adaptive sparse grid with 15 grid points (Navier-Stokes flow)

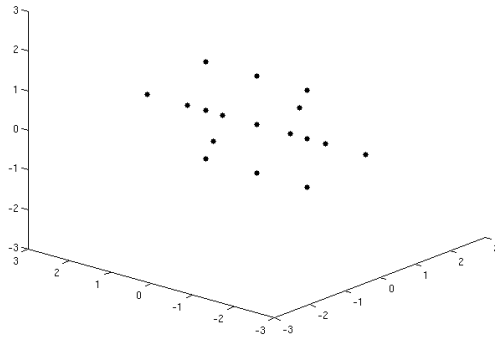


Fig. 22 Dimension adaptive sparse grid with 17 grid points (Navier-Stokes flow)

The linearity of the drag depending on the first eigenvector (cf. 18) is recognized by the dimension adaptive algorithm, so that this dimension is not further refined. Since the linear behaviour might change during the optimization, we add two points ensuring that the first eigenvector is taken into account during the optimization, see Fig. 22 Fig. 23 and 24 illustrate the drag and lift performance of the 17 perturbed geometries of the single setpoint and robust optimization.

The mean value of the drag in the robust case is a little bit higher than the mean value of the single setpoint optimization (5 drag counts) which can be attributed to the lift constraints. The robust optimization is feasible over the whole range of

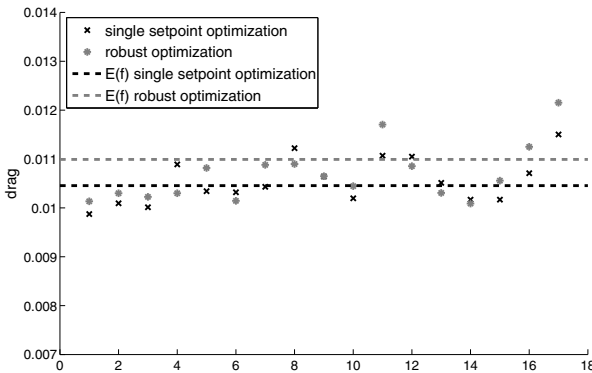


Fig. 23 Drag performance of the 17 perturbed geometries (Navier-Stokes)

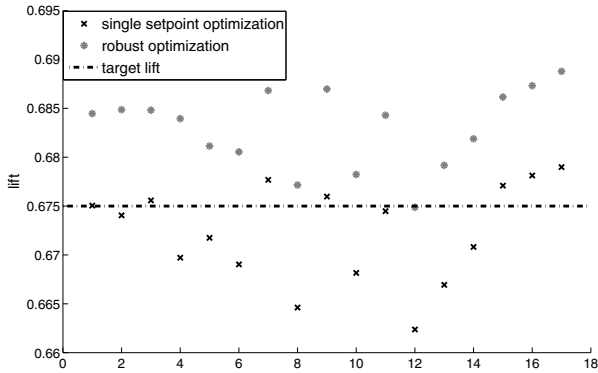


Fig. 24 Lift performance of the 17 perturbed geometries (Navier-Stokes)

perturbations as required, whereas the single setpoint optimization cannot reach the target lift in more than 60 % of the realizations. Hence, we can state that the semi-infinite optimizations leads to a better lift to drag ratio as in the Euler case. The resulting shapes are depicted in Fig. 25. The difference between the robust shape and the single setpoint optimized shape is smaller than in the Euler case, indicating that the profile is more sensitive to changes of the shape.

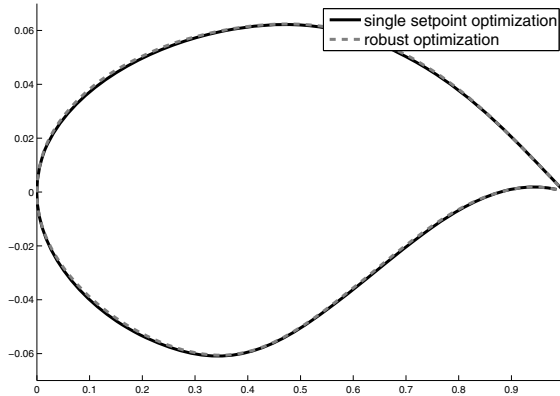


Fig. 25 Comparison of the optimized shapes (Navier-Stokes)

7.3 Numerical Study of the Influence of Geometrical Uncertainties (Testcase SFB-401)

The influence of geometrical uncertainties on flow parameters is investigated considering the 3d testcase SFB-401 (Navier-Stokes flow). A detailed overview and introduction to uncertainty quantification methods can be found in [42], [35], [24]. We will concentrate in the following on a Polynomial Chaos approach which expands the solution nonlinearly depending on the random variable s or random vector respectively, in a series of orthogonal polynomials with respect to the distribution of the random input vector

$$f(y, p, s(\zeta)) = \sum_{i=1}^{\infty} \alpha_i(y, p) \cdot \Phi_i(s(\zeta)) \quad (76)$$

with Φ_i orthogonal polynomials, $\alpha_i(y, p)$ deterministic coefficient functions. For a detailed discussion of the method, we refer to [41]. The basic idea of Polynomial Chaos representing the stochastic output of a differential equation with random input data is to reformulate the SDE replacing the solution and the right hand side of the PDE by a Polynomial Chaos expansion. Given a stochastic differential equation of the form

$$\mathcal{L}(x, t, \zeta; u) = g(x, t, \zeta), \quad x \in G, t \in [0, T], \zeta \in \Omega \quad (77)$$

where $u = u(x, t, \zeta)$ is the solution and $g(x, t, \zeta)$ is the right hand side. The operator \mathcal{L} can be nonlinear and appropriate initial and boundary conditions are assumed. Replacing pointwise the solution $u = u(x, t, \zeta)$ of equation (77) by the Polynomial Chaos expansion leads to

$$u(x, t, \zeta) \doteq \sum_{k=1}^M u_k(x, t) \Phi_k(s(\zeta)) \quad (78)$$

where $u_k(x, t)$ are the deterministic Polynomial Chaos coefficients which need to be determined. Furthermore, the right hand side $g(x, t, \zeta)$ will be also expanded by a Polynomial Chaos expansion

$$g(x, t, \zeta) = \sum_{k=1}^M g_k(x, t) \Phi_k(s(\zeta)) \quad (79)$$

and the deterministic coefficients are given by

$$g_k(x, t) = \frac{\langle g(x, t, \cdot) \Phi_k \rangle}{\langle \Phi_k^2 \rangle}. \quad (80)$$

In the literature, two different classes of methods determining the unknown coefficients $u_k(x, t)$ are proposed: non-intrusive and intrusive Polynomial Chaos. As the intrusive methods require a modification of the existing code solving the deterministic PDE (if the Operator \mathcal{L} is nonlinear), this drawback of the method force us

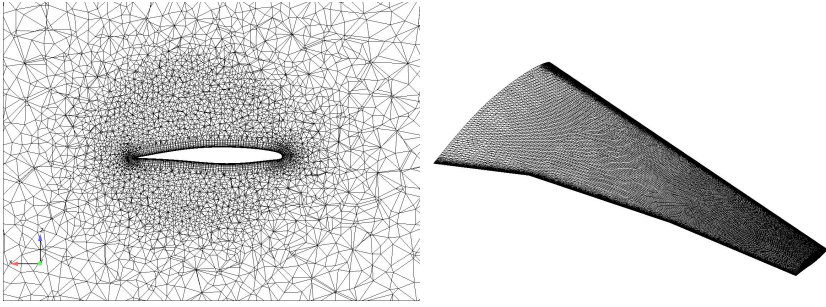


Fig. 26 Grid for the SFB-401 test case: cut of the grid at $y=0.5$ and surface of the wing

to consider non-intrusive Polynomial Chaos methods which allow to use the flow solver as a black-box. The unknown coefficients are given by

$$u_k(x,t) = \frac{\langle u(x,t,\cdot)\Phi_k \rangle}{\langle \Phi_k^2 \rangle} \tag{81}$$

$$= \frac{1}{\langle \Phi_k^2 \rangle} \int_{\mathbb{R}^d} u(x,t,v)\Phi_k(v)f_X(v)dv, \quad x \in G, t \in [0, T], \tag{82}$$

$$k = 1, \dots, M.$$

whereas the integral of equation (82) is approximated by a sparse grid approach.

In the 3d test case, the space is discretized by 2506637 grid points, where the surface is described by 80903 points. The grid generated with Centaur consists of 1636589 tetraeders, 4363281 prisms and 170706 surfacetriangles, 5427 surfacequadrilaterals, see Fig. 26. The geometrical uncertainties are assumed to occur on the upper side of the wing, the perturbed region is depicted in Fig. 27. The perturbations are modeled as a Gaussian random field with the following second order statistics

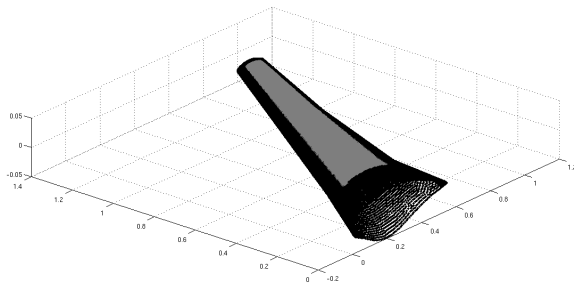


Fig. 27 Perturbed region of the SFB-401 profile

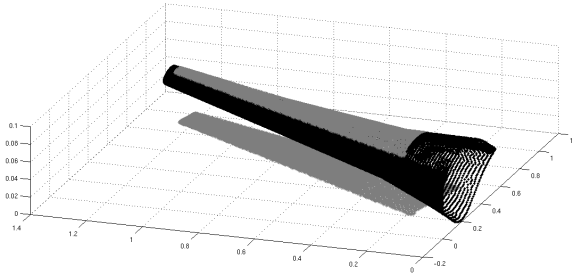


Fig. 28 Projection in 2d of the SFB-401 profile

$$\mathbb{E}(\psi(x, \zeta)) = 0 \quad \forall x \in \Gamma \tag{83}$$

$$Cov(x, y) = (0.0016)^2 \cdot \exp\left(-\frac{\|x-y\|^2}{(0.06)^2}\right) \quad \forall x, y \in \Gamma. \tag{84}$$

In order to take the curvature of the wing into account computing the norm in (84), the selected area is transformed into 2d approximating the distance by a polygon path on the surface. The projection is depicted in Fig. 28. Due to the problem size, an iterative eigensolver BLOPEX (cf. [28]) is used in order to solve the eigenvalue problem arising from the Karhunen-Loève expansion. The resulting eigenvalue distribution of the first 50 eigenvalues is shown in Fig. 29.

We consider the first 15 eigenvalues and eigenvectors to represent the random field, as an example the first, 8th and 15th eigenvectors and resulting perturbed shapes are depicted in Fig. 30.

In order to approximate statistics of the flow solution depending on the considered perturbations, the drag, the lift and the pressure coefficient c_p are expanded into the first 16 multidimensional Hermite polynomials. In the next two figures (Fig. 31, Fig. 32), the drag and the lift of each perturbed shape and the corresponding

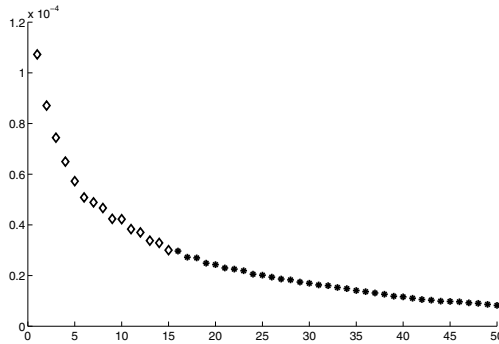


Fig. 29 Distribution of the first 50 eigenvalues of the given random field ψ

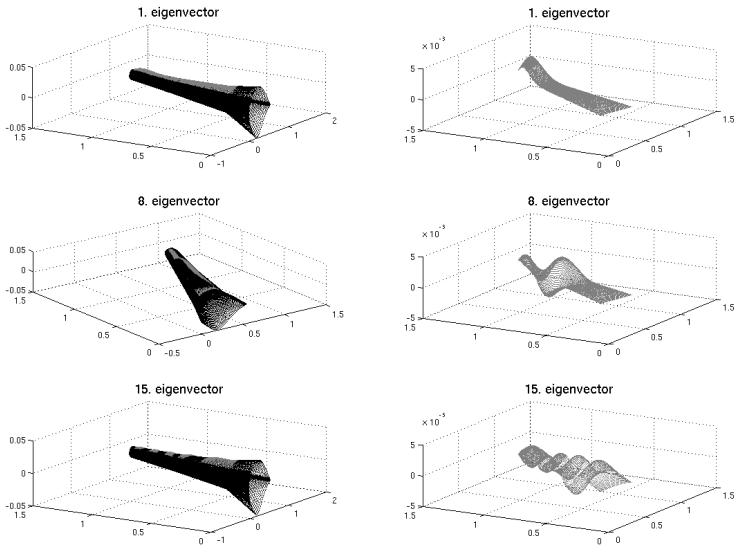


Fig. 30 First, 8th and 15th eigenvectors and resulting perturbed shapes

mean values are illustrated. As Fig. 31 indicates, the geometry uncertainties have a large impact on the target functional. The standard deviation from the mean value is equal to 1.65 drag counts, and the mean value is 6.55 drag counts higher than the nominal, unperturbed geometry. The comparison between the c_p distribution of the unperturbed geometry and the c_p distribution of the mean value shows that an

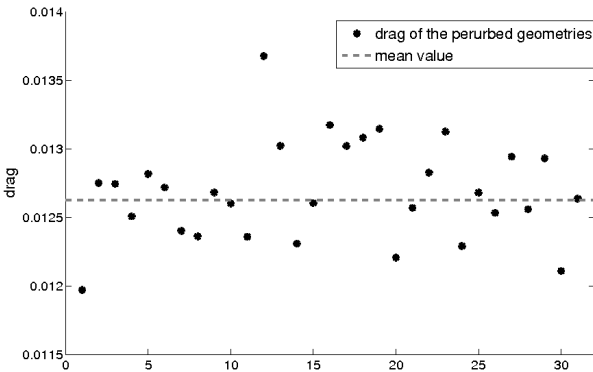


Fig. 31 Drag performance and mean value of the perturbed 3d shapes

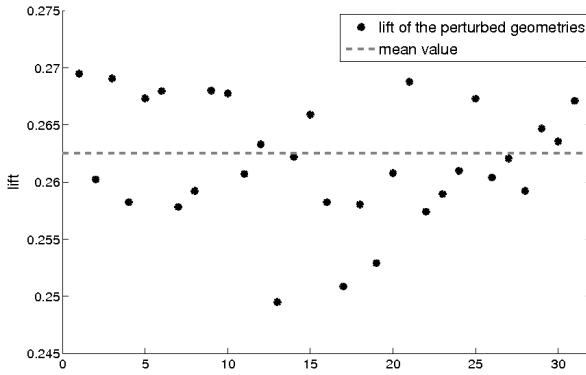


Fig. 32 Lift performance and mean value of the perturbed 3d shapes

additional shock on the upper side of the shape occurs due to the uncertainties in the geometry (cf. Fig. 34). Fig. 33 emphasizes the influence of the perturbations showing the variance of the c_p distribution.

8 Conclusions

Robust design is an important task to make aerodynamic shape optimization relevant for practical use. It is also highly challenging because the resulting optimiza-

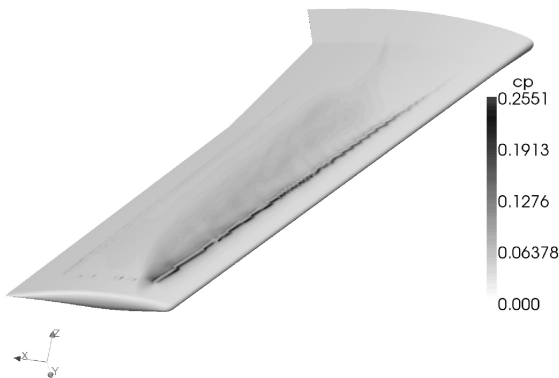


Fig. 33 Variance of the c_p distribution

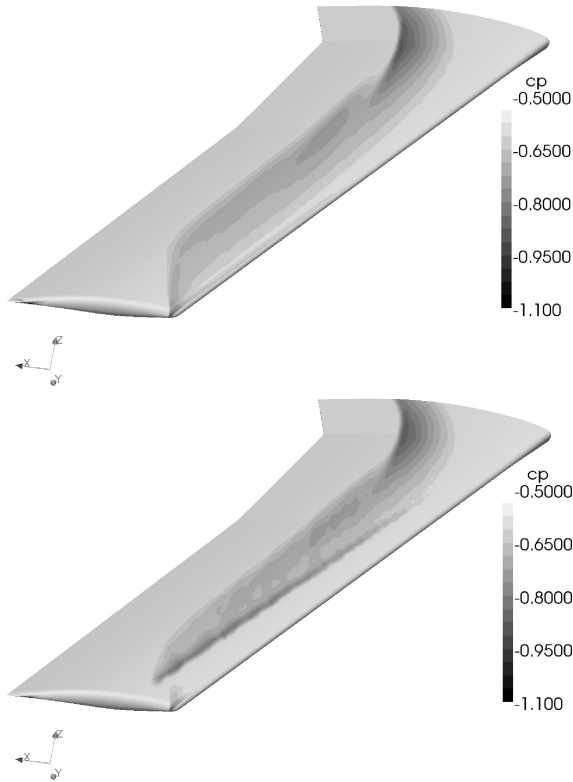


Fig. 34 Comparison between the c_p distribution of the unperturbed geometry (above) and the c_p distribution of the mean value (below)

tion tasks become much more complex than in the usual single set-point case. The comparison of the two robust optimization formulations shows that the discretized semi-infinite formulation seems to be of advantage in a numerical test case close to a real configuration. In the case of the geometric uncertainty, the approximation of the random field describing the perturbations of the geometry leads to a very high dimensional optimization task. The dimension of the probability space was efficiently reduced by a goal-oriented choice of the Karhunen-Loève basis. Furthermore, adaptive Sparse Grid techniques and one-shot methods have been successfully generalized to the semi-infinite formulation of the shape optimization problem in order to reduce the amount of computational effort in the resulting robust optimization. The numerical results show that even small deviations from the planned geometry have

a significant effect on the drag and lift coefficient, so that geometry uncertainties have to be taken into account in the aerodynamic design optimization problem to ensure a robust solution. The introduced methods can significantly reduce the costs of the robust optimization, so that robust design becomes numerically tractable in the aerodynamical framework.

References

- [1] Becker, R., Braack, M., Meidner, D., Rannacher, R., Vexler, B.: Adaptive finite element methods for pde-constrained optimal control problems. In: *Reactive Flows, Diffusion and Transport*, pp. 177–205 (2006)
- [2] Bock, H.G., Egartner, W., Kappis, W., Schulz, V.: Practical shape optimization for turbine and compressor blades by the use of PRSQP methods. *Optimization and Engineering* 3(4), 395–414 (2002)
- [3] Bock, H.G., Kostina, E., Schäfer, A., Schlöder, J.P., Schulz, V.H.: Multiple set point partially reduced SQP method for optimal control of PDE. in reactive flows, diffusion and transport. In: Jäger, W., Rannacher, R., Warnatz, J. (eds.) *Reactive Flows, Diffusion and Transport. From Experiments via Mathematical Modeling to Numerical Simulation and Optimization Final Report of SFB (Collaborative Research Center) 359* (2007)
- [4] Bock, H.G., Schulz, V.: Mathematical aspects of CFD-model based optimization. In: Thevenin, D., Janiga, G. (eds.) *Optimization and Computational Fluid Dynamics*, pp. 61–78. Springer (2007)
- [5] Borzi, A., von Winkel, G.: Multigrid methods and sparse-grid collocation techniques for parabolic optimal control problems with random coefficients. *SIAM Journal on Scientific Computing* 31(3), 2172–2192 (2009)
- [6] Bungartz, H.-J.: *Finite Elements of Higher Order on Sparse Grids*. Habilitationsschrift, Fakultät für Informatik, Technische Universität München, Aachen (November 1998)
- [7] Bungartz, H.-J., Dirnstorfer, S.: Multivariate quadrature on adaptive sparse grids. *Computing* 71(1), 89–114 (2003)
- [8] Davis, P.J., Rabinowitz, P.: *Methods of Numerical Integration*, 2nd edn. Dover Publications (November 2007)
- [9] Eldred, M.S., Webster, C.G., Constantine, P.: *Evaluation of non-intrusive approaches for wiener-asky generalized polynomial chaos* (2008)
- [10] Christodoulos, A., Floudas, C.A., Stein, O.: The adaptive convexification algorithm: A feasible point method for semi-infinite programming. *SIAM J. on Optimization* 18(4), 1187–1208 (2007)
- [11] Frauenfelder, P., Schwab, C., Todor, R.A.: Finite elements for elliptic problems with stochastic coefficients. *Computer Methods in Applied Mechanics and Engineering* 194(2-5), 205–228 (2005); Selected papers from the 11th Conference on the Mathematics of Finite Elements and Applications
- [12] Garcke, J., Griebel, M., Thess, M.: Data mining with sparse grids. *Computing* 67(3), 225–253 (2001)
- [13] Garcke, J.: A dimension adaptive sparse grid combination technique for machine learning. In: Read, W., Larson, J.W., Roberts, A.J. (eds.) *Proceedings of the 13th Biennial Computational Techniques and Applications Conference, CTAC 2006*. ANZIAM J., vol. 48, pp. C725–C740 (2007)

- [14] Gerstner, T., Griebel, M.: Dimension-adaptive tensor-product quadrature. *Computing* 71(1), 65–87 (2003)
- [15] Ghanem, R.G., Spanos, P.D.: *Stochastic Finite Elements: A Spectral Approach*. Dover Publ. Inc. (2003)
- [16] Gherman, I.: Approximate Partially Reduced SQP Approaches for Aerodynamic Shape Optimization Problems. PhD thesis, University of Trier, Trier, Germany (2008)
- [17] Gumbert, C.R., Newman, P.A., Hou, G.J.-W.: High-fidelity computational optimization for 3-d flexible wings: Part ii effect of random geometric uncertainty on design. *Optimization and Engineering* 6(1), 139–156 (2005)
- [18] Hazra, S.B., Schulz, V.: Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints. *SIAM J. Sci. Comput.* 28(3), 1078–1099 (2006)
- [19] Hazra, S.B., Schulz, V., Brezillon, J., Gauger, N.R.: Aerodynamic shape optimization using simultaneous pseudo-timestepping. *J. Comput. Phys.* 204(1), 46–64 (2005)
- [20] Henrion, R.: Structural properties of linear probabilistic constraints. *Optimization* 56(16), 425–440 (2007)
- [21] Huyse, L., Lewis, R.M., Li, W., Padula, S.L.: Probabilistic approach to free-form airfoil shape optimization under uncertainty. *AIAA Journal* 40, 1764–1772 (2002)
- [22] Kall, P., Wallace, S.W.: *Stochastic Programming*. Wiley (1994)
- [23] Karhunen, K.: Zur Spektraltheorie stochastischer Prozesse. *Suomalaisen Tiedeakatemian Toimituksia: Ser. A: 1, Mathematica, Physica* 34 (1946)
- [24] Keese, A.: A review of recent developments in the numerical solution of stochastic pdes (stochastic finite elements). Technical Report 2003-6, Technische Universitt Braunschweig, Brunswick (2003)
- [25] Khoromskij, B.N., Litvinenko, A., Matthies, H.G.: Application of hierarchical matrices for computing the karhunen-loève expansion. *Computing* 84(1-2), 49–67 (2009)
- [26] Klimke, A.: Uncertainty modeling using fuzzy arithmetic and sparse grids. PhD thesis, Universität Stuttgart, Shaker Verlag, Aachen (2006)
- [27] Klimke, A., Wohlmuth, B.: Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in matlab. *ACM Trans. Math. Softw.* 31(4), 561–579 (2005)
- [28] Knyazev, A.V.: Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM J. Sci. Comput.* 23, 517–541 (2001)
- [29] Li, W., Huyse, L., Padula, S.: Robust airfoil optimization to achieve drag reduction over a range of mach numbers. *Structural and Multidisciplinary Optimization* 24(1), 38–50 (2002)
- [30] Li, W., Padula, S.L.: Robust airfoil optimization in high resolution design space. ICASE NASA Langley Research Centre (2003)
- [31] Loève, M.: *Probability Theory* 1, 4th edn. Springer (1994)
- [32] Loeven, G.J.A., Bijl, H.: Airfoil analysis with uncertain geometry using the probabilistic collocation method. In: 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA 2008-2070 (2008)
- [33] Ma, X., Zabarav, N.: An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.* 228(8), 3084–3113 (2009)
- [34] Mercer, J.: Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London* (1909)
- [35] Najm, H.N.: Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics* 41(1) (2009)

- [36] Padula, S., Gumbert, C., Li, W.: Aerospace applications of optimization under uncertainty. In: ISUMA 2003: Proceedings of the 4th International Symposium on Uncertainty Modelling and Analysis, p. 286. IEEE Computer Society, Washington, DC (2003)
- [37] Prékopa, A.: Stochastic Programming. Kluwer Academic Publishers, Dordrecht (1995)
- [38] Putko, M., Newman, P., Taylor III, A., Green, L.: Approach for uncertainty propagation and robust design in cfd using sensitivity derivatives. In: Design in CFD Using Sensitivity Derivatives, AIAA Paper 2001-2528, in AIAA 15th Computational Fluid Dynamics Conference, pp. 2001–2528 (2001)
- [39] Schulz, V., Schillings, C.: On the nature and treatment of uncertainties in aerodynamic design. *AIAA Journal* 47(3), 646–654 (2009)
- [40] Smolyak, S.A.: Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics Doklady* 4, 240–243 (1963)
- [41] Xiu, D., Karniadakis, G.E.: Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of Computational Physics* 187(1), 137–167 (2003)
- [42] Zang, T.A., Hensch, M.J., Hilburger, M.W., Kenny, S.P., Luckring, J.M., Maghami, P., Padula, S.L., Jefferson Strout, W.: Needs and opportunities for uncertainty-based multidisciplinary design methods for aerospace vehicles (2002)
- [43] Zhang, Y.: A general robust-optimization formulation for nonlinear programming. *J. Optim. Theory Appl.* (2004)

Author Index

- Albensoeder, S. 55
- Ballmann, Josef 181
- Barnewitz, Holger 219
- Behr, Marek 181
- Bogdanski, S. 247
- Filimon, Alexander 129
- Fliester, Horst 181
- Gauger Nicolas R. 29
- Haupt, M. 157
- Henes, D. 247
- Horst, P. 157
- Krämer, E. 247
- Litvinenko, Alexander 265
- Liu, Dishu 283
- Lutz, Th. 247
- Matthies, Hermann G. 265
- Mazlum, E. 3
- Meinke, Matthias 101
- Mockett, Charles 77
- Munz, Claus-Dieter 129
- Orlt, Matthias 29
- Radespiel, R. 3
- Reich, P. 157
- Reim, A. 157
- Reimer, Lars 181
- Roidl, Benedikt 101
- Schillings, Claudia 297
- Schmidt, Tobias 77
- Schröder, Wolfgang 101
- Schulz, Volker 297
- Stickan, Bernd 219
- Thiele, Frank 77
- Wellmer, Georg 181
- Wolf, A. 247