Aerodynamic Topology Optimisation Using an Implicit Representation and a Multiobjective Genetic Algorithm

Windo Hutabarat, Geoffrey T. Parks, Jerome P. Jarrett, William N. Dawes, and P. John Clarkson

Engineering Design Centre, Department of Engineering
University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK
{wh226,gtp,jpj1001,wnd,pjc10}@eng.cam.ac.uk

Abstract. Given the focus on incremental change in existing empirical aerodynamic design methods, radical, unintuitive, new optimal solutions in previously unexplored regions of design space are very unlikely to be found using them. We present a framework based on an implicit shape representation and a multiobjective evolutionary algorithm that aims to produce a variety of optimal flow topologies for a given requirement, providing designers with insights into possibly radical solutions. A revolutionary integrated flow simulation system developed specifically for design work is used to evaluate candidate designs.

1 Introduction

Fluid dynamics is a complex and nonlinear discipline. Predicting the behaviour of aerodynamic objects is not easy. Hence, aerodynamic design processes are rarely started from first principles. Initial decisions in aerodynamic design are usually based on empirical knowledge [1]. However, it can be argued that, in some areas, aerodynamic designers have approached the point where using an overly derivative approach may lead to some basic designs being used outside of their optimal region. We shall illustrate this point with an example.

The Grid Fin Example. The wing is the solution most commonly used whenever the generation of lift is required. By continuous incremental improvement to its basic shape and working principle, there are now many derivatives of wings which are widely applicable. However, today's increasingly demanding design requirements may uncover situations in which none of these variations is optimal; indeed, the situation may be such that a radically different shape is necessary.

Figure 1 shows an advanced air-to-air missile equipped with a novel type of lifting surface called *the grid fin.* This trellis-like contraption at the tail of the missile has characteristics that happen to be very well suited to the demanding requirements of supersonic dogfighting [2]. It is clear that using grid fins instead of wing derivatives contributes to the fact that this particular missile (the Vympel R-77) is currently widely held to be the premier supersonic dogfighting missile.

N. Monmarché et al. (Eds.): EA 2007, LNCS 4926, pp. 148–159, 2008.

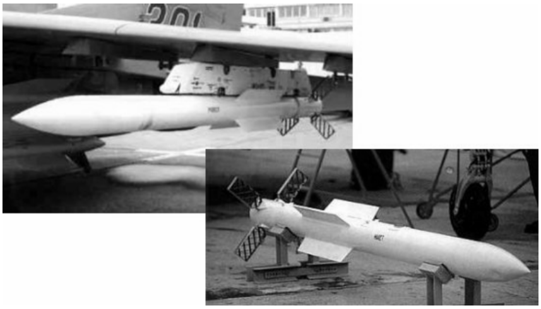

Fig. 1. Grid fins on the Vympel R-77 missile [**?**]

1.1 Topology as a Design Variable

The difference between the wing and the grid fin is greater than can be captured by the concept of *shape*; they are said to be different in *topology*. Due to the high cost of change, it is important to select the "right" topology early in the design process. Ideally, therefore, topology should be one of the key variables determined during the conceptual design phase. However, the authors are not aware of any method or tool that designers can use early in the design process to explore alternative aerodynamic topologies. The topology of a design is therefore usually predetermined before the design process even begins.

One possible solution is suggested by the use of simulation-based multiobjective optimisation to produce optimal designs and parameter trade-offs [4,5]. If such a system can be used to explore the topology trade-off, its output may help designers gain insights into radical and previously unconsidered options.

1.2 Proposed Framework

We aim to demonstrate that, for a given set of requirements, a framework consisting of a *stochastic, multiobjective optimiser* using a *topologically unconstrained shape representation* and coupled with a *robust CFD evaluator* is able to consistently identify a variety of solution flow topologies, which will hopefully provide designers with insight into the available topological trade-offs.

A stochastic optimiser is widely recognised to have the following characteristics:

1. The ability to incorporate practically any sort of design objectives;
2. The ability to treat evaluation codes as black-boxes;
3. The ability to escape local optima;
4. Easy parallelisation.

We consider these characteristics to be more suitable in the information-starved, early stages of a design process than alternative approaches, despite the usual perceived disadvantage of this choice: that of demanding a large number of objective function evaluations.

We have demonstrated the viability of a basic version of this proposed framework in an earlier study [6]. Our current work extends this concept to introduce multiobjective optimisation, continuous surfaces, and a proper CFD code into the framework.

The availability of a suitable CFD flow solver is absolutely crucial. We are grateful to Cambridge Flow Solutions for providing us with BoXeR, a newly-developed CFD package with exactly the sort of capabilities that we need [7].

2 Related Work

Shape optimisation in fluids has been the subject of intense research [8], and there are various ways in which CFD can be used for this purpose [9]; combining CFD with stochastic optimisation algorithms has proven to be quite a successful approach [5].

Nevertheless, research in topology optimisation in fluids has only started very recently with the pioneering work of Borrvall and Petersson [10]. This work is based on the material distribution approach, a well-known method of structural topology optimisation [11]. The state of the art of this approach is summarised in Gersborg-Hansen's thesis [12]. As yet, however, this approach has not been demonstrated using a finite-volume Navier-Stokes discretisation, which is the best established approach in CFD.

Another approach [13] can be viewed as an extension of the Evolutionary Structural Optimization method [14]. This approach adds or removes Boolean cells according to a set of flow-based optimality criteria.

Our approach is built on previous work in Genetic Algorithm (GA)-based structural topology optimisation. Our previous work [6] can be seen as an extension of voxel-based GA structural topology optimisation work [15], while the present work can be viewed as an extension of Kita and Tanie's work [16].

One topologically unconstrained way of representing shapes is by using an implicit representation. Examples of shape optimisation using implicit shape representation are provided by the level-set community, such as [17].

The Radial Basis Function (RBF) has been recognised as an efficient way of storing implicit representations. It has been shown to be compatible with topological optimisation [18]. One of the chief objections to implicit representations is that they cannot represent complex details without running foul of the curse of dimensionality. However, methods exist that enable an RBF representation to represent shapes of nearly arbitrary complexity [19].

The development of BoXeR was first reported in [20]. Since it is not yet commercially deployed, further information may be found on the Cambridge Flow Solutions website [7].

3 Framework Implementation

3.1 Topologically Unconstrained Shape Representation

The literature on topology optimisation suggests three main topologically unconstrained representation methods:

1. Binary occupancy: [15] in structures and [13] in fluids
2. Material distribution: [11] in structures and [10] in fluids
3. Implicit representation: [17] in structures and the present work in fluids

In our previous work [6] we used a simple binary representation scheme, which unfortunately is too expensive to use to model continuous surfaces. The second method, material distribution, requires very extensive modification of the flow simulation, which is not a realistic option.

In the present work we use an implicit representation with information stored in an RBF equation. In an implicit representation, an object Ω with boundary ω is defined as the set of points $\mathbf{x}$

$$\{\mathbf{x} : s(\mathbf{x}) = p\}, \ \mathbf{x} \in \omega \tag{1}$$

where p can be set to any scalar value; usually $p = 0$.

RBFs have been used by several research groups as representation methods of solids and surface interpolators [21]. An RBF interpolation process constructs an implicit function $s(\mathbf{x})$ by using a set of control point coordinates $\mathbf{x}_i$ and function value s_i at $\mathbf{x}_i$. We shall briefly describe our RBF implementation.

RBF Construction. An RBF is basically a weighted sum of a given basis function that is evaluated over the distances of all pairs of a set of control points. More elaborately, an implicit function $s(\mathbf{x})$ is expressed as

$$s(\mathbf{x}) = \sum_{i=1}^{N} \lambda_i \phi(|\mathbf{x} - \mathbf{x}_i|), \ \mathbf{x} \in \mathbf{R}^d \tag{2}$$

where $\mathbf{x}_i$ are the locations of the control points, N is the number of control points, λ_i is the weight for control point $\mathbf{x}_i$, $\phi(r_i)$ is the basis function, and $||$ is the Euclidean norm in $\mathbf{R}^d$. The basis function $\phi(r_i)$ is usually chosen from a family of well-known functions, such as the thin-plate, Gaussian, multiquadric, etc. The multiquadric basis function

$$\phi(r_i) = \sqrt{\epsilon \cdot r_i^2 + 1} \tag{3}$$

is quite widely used. Here ϵ is an adjustable constant.

The RBF $s(\mathbf{x})$ is completed by calculating the weights λ_i. If we define

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{11} & \phi_{12} & \cdots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \cdots & \phi_{2N} \\ \vdots & \vdots & & \vdots \\ \phi_{N1} & \phi_{N2} & \cdots & \phi_{NN} \end{bmatrix}, \text{ where } \phi_{ij} = \phi(|\mathbf{x}_i - \mathbf{x}_j|) \tag{4}$$

$$\mathbf{\Lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N)^T \tag{5}$$

$$\mathbf{S} = (s_1, s_2, \ldots, s_N)^T \tag{6}$$

then the approximation process is performed by solving the linear system of equations:

$$\mathbf{\Phi} \cdot \mathbf{\Lambda} = \mathbf{S} \tag{7}$$

The function $s(\mathbf{x})$ can then be used as an implicit shape representation.

Initial solutions can be generated by generating control points – either randomly or uniformly – and then giving each control point a random scalar value. The embedded shape can be manipulated either by changing the values of the control point scalars or by addition/removal of control points. Should we choose to change only the scalar values and fix the number and locations of the control points, we can then use standard real-valued genetic search operators. Prior to shape evaluation, the embedded shape can be extracted from the implicit function by a variety of methods, such as the exhaustive Marching Cubes method.

In the present work, we chose to limit ourselves to 2D implicit surfaces, with the control points spread in a regular 2D lattice. The embedded shapes are edited by changing the nodal scalar values. The shape generation method then resembles the generation of an elevation map, and shape manipulation changing the elevation of some nodes. The steps required to create a shape, shown schematically in Figure 2, are as follows:

(a) A regular $I \times J$ lattice of N control points is created.
(b) Each control point is associated with a scalar value s, where $s_{\min} < s < s_{\max}$. In Figure 2(b) the scalar value is plotted as a height map.
(c) A RBF implicit function is created according to the control points.
(d) The embedded shape can be extracted as an isoline of a specified value $s = s_\omega$.

In most cases, the generated isolines will not form closed shapes, intersecting the shape generation boundary instead. We can choose whether to leave these

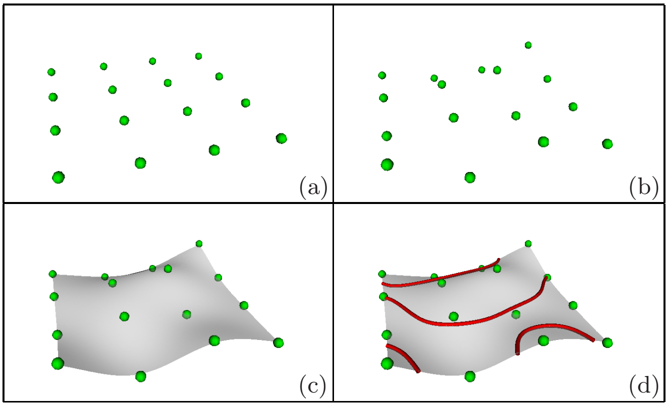

Fig. 2. Implicit representation of a 2D object. The end result of this process is the embedded shape, shown as the red line in (d). Prior to evaluation, the red lines will be extruded into surfaces as shown in Figure 4.

shapes open or to close them with some repair mechanism. In the present implementation we opt for allowing non-closed shapes; this has some important consequences that will be encountered later.

3.2 Optimisation Algorithm

For the optimisation algorithm we chose the well-known multiobjective GA NSGA-II [22] due to its demonstrated capability in flow geometry optimisation [23] and in unveiling innovative design principles [4].

Design Variables. The design vector **s** is the vector of scalar values stored in each interpolation node. These are standard real-valued design variables; hence the special crossover and mutation operators that we originally proposed [6] are currently not necessary. We limit the scalar values of the control points to the range $-1 < s < 1$, and we extract the shape boundary at $s = 0$.

Genetic Operators. Since no special operators are necessary, the crossover and mutation operators can be chosen from the wide range of real-valued GA operators available in the literature. With ease of implementation as our main criterion, we chose Parent-centric Normal Crossover (PNX) [24] and the standard Gaussian or "normal" mutation operator [25] in preference to the simulated binary crossover (SBX) and polynomial mutation operators suggested in the original NSGA-II paper [22]. We have used these operators with our initial test cases and found their performance to be satisfactory.

3.3 Evaluation

BoXeR represents a new approach to the use of flow simulation in the design process. Recognising the bottlenecks of CFD usage, Cambridge Flow Solutions wraps revolutionary features around their industry-standard "NEWT" code, a finite-volume Navier-Stokes flow solver, producing an integrated and parallel geometry kernel, mesh generator, Cartesian flow solver, and post-processor [7]. Among the revolutionary features of BoXeR, our work benefited most from its automated mesh generation capability.

The success of discretised flow simulation is highly dependent on the quality of its mesh. CFD meshes are usually built starting from existing CAD models, which are rarely designed with CFD use in mind [26]. This makes the process laborious and error-prone, quite unsuited to an automated optimisation system. BoXeR's integrated CAD-importing and mesh generation tools do away with this problem. The BoXeR user can simply define the domain, the geometry, and the flow conditions, then step back and watch as the system automatically meshes, refines, and iterates towards convergence in real-time.

4 Test Case

In this test case we set up a simple optimisation problem and see if our framework achieves what we seek to demonstrate. We first put the shape generation box

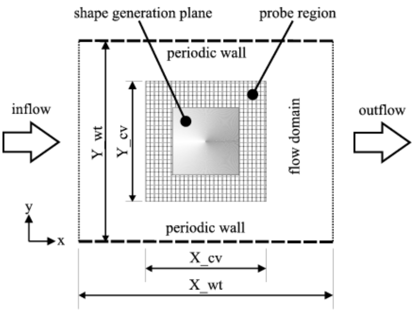

Fig. 3. Virtual wind tunnel dimensions

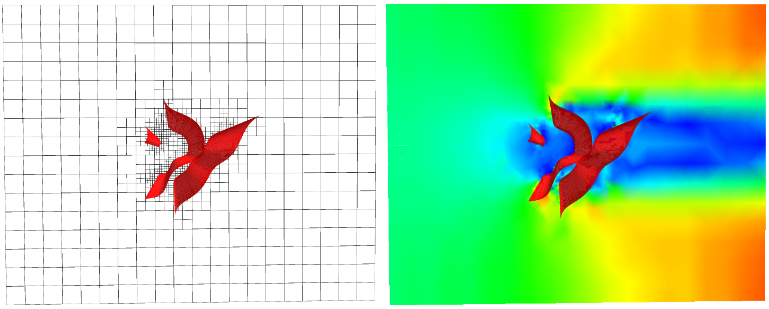

Fig. 4. BoXeR octree domain discretisation, and a Mach number colour-mapping visualisation of the simulated flow. The curvy surfaces in the middle is the extruded shape.

into a BoXeR virtual wind tunnel, as shown in Figure 3. BoXeR is then run for a given number of iterations, and the resulting flow field is processed to extract the objective function values. We wish to see whether:

1. The framework is consistently able to produce Pareto fronts.
2. The resulting Pareto front is composed of a variety of flow topologies.
3. The resulting shapes are "aesthetically pleasing". We shall use this catch-all clause to direct our next efforts.

Tables 1 and 2 show the parameters used in the GA and the virtual wind tunnel, respectively. The GA parameters in Table 1 are those suggested in [22], which

Table 1. Optimisation parameters

Number of control points	$I = J = 8$	Basis function	Multiquadric
Control point scalar bounds	$s = [-1.0, 1.0]$	Shape boundary value	$s_\omega = 0.0$
Number of individuals	100	Crossover rate	0.85
Mutation rate	0.033	Number of generations	100

Table 2. BoXeR wind tunnel parameters

Wind tunnel dimensions	$\mathrm{X_{wt}} = 10\,\mathrm{m}, \mathrm{Y_{wt}} = 8\,\mathrm{m}, \mathrm{Z_{wt}} = 1\,\mathrm{m}$		
Control volume dimensions	$\mathrm{X_{cv}} = 5\,\mathrm{m}, \mathrm{Y_{cv}} = 4\,\mathrm{m}, \mathrm{Z_{cv}} = 0.5\,\mathrm{m}$		
Test generation plane dimensions	$\mathrm{X_g} = \mathrm{Y_g} = 1\,\mathrm{m}$		
Specific heat capacity	$c_p = 1005\,\mathrm{J\,kg^{-1}K^{-1}}$	Heat capacity ratio	$\gamma = 1.4$
Kinematic viscosity	$\mu = 1 \times 10^{-5}\mathrm{m^2s^{-1}}$	Inlet total pressure	$1 \times 10^5\,\mathrm{Pa}$
Inlet total temperature	$288\,\mathrm{K}$	Inlet static pressure	$7.56 \times 10^4\,\mathrm{Pa}$

were found to be satisfactory after a small number of test runs. In Table 2, the dimensions of the virtual wind tunnel are chosen to model a comfortably large wind tunnel with a 2 m by 2 m test section. The last six parameters in Table 2 define the flow to be under atmospheric, subsonic conditions, a regime of great interest but free of the complications found in transonic or supersonic flow.

4.1 Objective Functions

The design objective is simple: given a uniform, horizontal incoming fluid flow, what kind of topology will produce maximum upward momentum with the least loss in horizontal momentum? This crudely approximates the lift-drag trade-off of a downforce generator on a racing car.

Translating this to the optimisation framework, the objective evaluator is then a post-processor that integrates the momentum fluxes within a control volume $\mathbf{p}_{cv}$. We define the control volume as a 3D hexahedron centrally surrounding the shape generation plane, having dimensions exactly half the corresponding dimensions of the simulation domain. The momentum integration is

$$\mathbf{p}_{cv} = m_{cv} \cdot \mathbf{u}_{cv} = \sum_{i=1}^{N_{cv}} \rho_i \cdot V_i \cdot \mathbf{u}_i \tag{8}$$

where N_{cv} is the number of cells within the control volume, V_i is the volume of cell i, and ρ_i and $\mathbf{u}_i$ are the density and velocity of the fluid in that cell.

A large vertical momentum means that the structure performs well in deflecting flow upwards, while a large horizontal momentum means that the design creates minimal drag. Hence, if a given design variable vector $\mathbf{v}$ produces $\mathbf{p}_{cv} = (p_x, p_y)^T_{cv}$, then the two objectives to be *minimised* are

$$f_{objective_1}(\mathbf{v}) = -p_x \tag{9}$$

$$f_{objective_2}(\mathbf{v}) = -p_y \tag{10}$$

4.2 Selection Operator

To maintain population diversity, NSGA-II applications typically use a crowded tournament selection operator working in objective function space. We use a design-space-based crowding operator instead, since our initial experiments suggest that, while this operator causes the system converge slower, the final population usually has greater topological variety.

5 Results and Discussion

Figure 5 shows the entire population of the final generation of a typical test case run. This entails 10,000 separate runs of the CFD code with 2000 iterations towards convergence each. Each test case run takes approximately 24 hours to run on a cluster of twelve Opteron nodes.

Figure 6 shows a selection of shapes that result from one optimisation run. It is important to remember that in aerodynamic shape optimisation the objective function is extracted not from the shape of the material but from the shape of the void, or, more accurately, from the flow that is affected by the shape of the void regions.

We see here a collection of shapes that importantly create flows with different topological characteristics. The results range from the singular shape that performs minimal flow manipulation, to a strongly curved shape that has more effect on the flow, to multiple curved shapes that multiply the effect of the singular curved shape. In short, we see that the optimisation system has created varying topologies generating flow patterns that minimise the two conflicting objectives to varying degrees.

The shapes still have quirky undulations which may be smoothed if we continue the optimisation further, but we feel that, at this stage, it is more important to prove that the system can find "embryonic" solution topologies.

Apparently, our allowing non-closed shapes has resulted in a final population consisting of aesthetically unpleasing plate-like shapes. Apart from the obvious (but not modeled) structural problems associated with such structures, the optimiser misses out on the opportunity of finer flow control useful in the latter stages of the optimisation, since it has no separate control of the two sides of material in contact with the flow. This, however, suggests a hybrid optimisation approach, where these results are then used as the initial points for a more conventional shape optimisation.

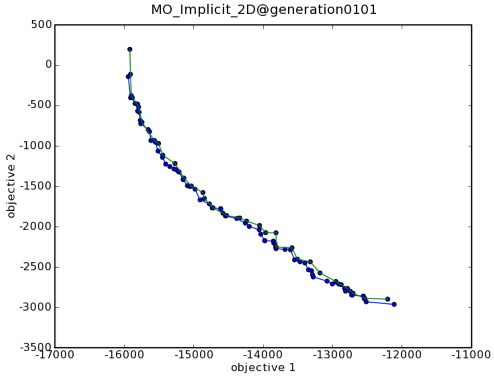

Fig. 5. A typical Pareto front from the test case The lines connects solutions belonging to the same Pareto rank as defined by NSGA-II, this particular one has two ranks

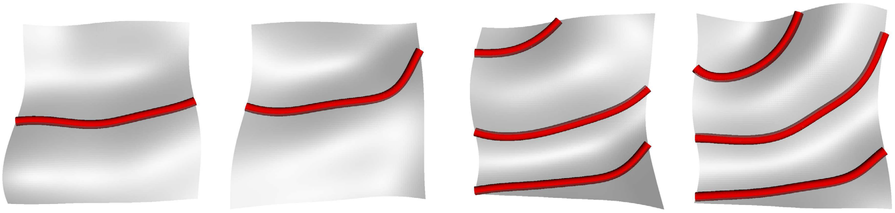

Fig. 6. Some samples from the Pareto front. The fluid flows from left to right, and is deflected upward. The carpet-like surfaces directly facing the viewer are the visualisation of the RBF interpolation surface, the thick lines represents s_ω isolines. Extruded surface is omitted for clarity. The objective function values are, respectively: $(-15769, -685)$, $(-14610, -1779)$, $(-13963, -2075)$, and $(-12556, -2862)$.

Comparing the results with our list of objectives given in Section 4, we found that in the limited number of tests we have done, the framework is consistently able to produce the Pareto front for the given test case. Our first objective is thus achieved.

As can be seen from Figure 6, the Pareto front consists of shapes with variable topologies, albeit non-closed. This demonstrates the ability of our framework to produce a trade-off across flow topologies. This means that the second objective is also achieved.

However, with respect to the third objective, we found the resulting non-closed shapes aesthetically unpleasing. This suggests that the optimisation problem can be better defined to produce aesthetically more pleasing closed shapes. One way to tackle this is by placing a constraint on non-closed shapes. We can do this by exploiting the fact that non-closed shapes will have a sign change on the interpolated scalar field boundaries. The extent of these scalar sign changes can be used as a measure of the extent of constraint violation.

6 Conclusions and Future Work

Our previous work has shown that GAs can be used to perform fluidic topology optimisation. Our present work seeks to demonstrate that a multiobjective GA, driving a topologically unconstrained representation, coupled with a proper CFD code, is able to come up with a solution topology trade-off. Most of our objectives have been achieved, with the remaining objective providing the impetus for further improvement.

In the future, we aim to do the following:

– Implement an objective function that extracts the non-dimensional aerodynamic characteristics of a shape. This will provide a comparative figure of merit that is better and more familiar to aerodynamic designers.
– Study the effects of the dimensions of the virtual wind tunnel.
– Further exploration of the robustness of the GA operator parameters.

Acknowledgments

The support of Cambridge Flow Solutions and of the UK Engineering and Physical Sciences Research Council (EPSRC) under grants GR/R64100/01 and EP/E001777/1 is gratefully acknowledged.

References

1. Giles, M.B.: Aerospace Design: a Complex Task. Technical Report 97/07, Oxford University Computing Laboratory (1997)
2. Fluent Inc.: First Viscous Analysis of Grid Fins Gives Better Prediction of Missile Trajectory. JA136 Journal Articles by Fluent Software Users, Fluent Inc. (2001)
3. Scott, J.: Missile Grid Fin (2006) (Last checked on September 15th, 2007), `http://www.aerospaceweb.org`
4. Deb, K.: Unveiling Innovative Design Principles by Means of Multiple Conflicting Objectives. Engineering Optimization 35(5), 445–470 (2003)
5. Sasaki, D., Obayashi, S., Nakahashi, K.: Navier-Stokes Optimization of Supersonic Wings with Four Objectives Using Evolutionary Algorithm. Journal of Aircraft 39 (2002)
6. Hutabarat, W., Parks, G.T., Jarrett, J.P., Clarkson, P.J.: A New Approach to Aerodynamic Topology Optimization. In: Parmee, I. (ed.) Adaptive Computing in Design and Manufacture, vol. 2006 (2006)
7. Cambridge Flow Solutions: BoXeR (2007) (Last checked on September 15th, 2007), `http://www.cambridgeflowsolutions.com`
8. Mohammadi, B., Pironneau, O.: Applied Shape Optimization for Fluids. Oxford University Press, Oxford (2001)
9. Jameson, A., Vassberg, J.C.: Computational Fluid Dynamics for Aerodynamic Design: Its Current and Future Impact. In: 39th AIAA Aerospace Sciences Meeting and Exhibit, AIAA (2001)
10. Borrvall, T., Petersson, J.: Topology Optimization of Fluids in Stokes Flow. International Journal for Numerical Methods in Fluids 41, 77–107 (2003)
11. Bendsoe, M., Kikuchi, N.: Generating Optimal Topologies in Structural Design Using a Homogenization Method. Computer Methods in Applied Mechanics and Engineering 71(2), 197–224 (1988)
12. Gersborg-Hansen, A.: Topology Optimization of Flow Problems. PhD thesis, Technical University of Denmark (2007)
13. Häussler, P., Nitsopoulos, I., Sauter, J., Stephan, M.: Topology and Shape Optimization Methods for CFD Problems. In: 24th CADFEM Users' Meeting 2006, International Congress on FEM Technology (2006)
14. Xie, Y., Steven, G.: Evolutionary Structural Optimization. Springer, Heidelberg (1997)
15. Kane, C., Schoenauer, M.: Topological Optimum Design Using Genetic Algorithms. Control and Cybernetics 25(5), 1059–1088 (1996)
16. Kita, E., Tamaki, T., Tanie, H.: Topology and Shape Optimization of Continuum Structures by Genetic Algorithm and BEM. Computer Assisted Mechanics and Engineering Sciences 11, 63–75 (2004)
17. Allaire, G., Gournay, F.D., Jouve, F., Toader, A.M.: Structural Optimization Using Topological and Shape Sensitivity Via a Level Set Method. Technical report, Ecole Polytechnique (2004)

18. Wang, S., Wang, M.: Radial Basis Functions and Level Set Method for Structural Topology Optimization. International Journal for Numerical Methods in Engineering 65(12), 2060–2090 (2005)
19. Carr, J., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B., Evans, T.: Reconstruction and Representation of 3D Objects with Radial Basis Functions. In: Conference on Computer Graphics and Interactive Techniques, pp. 67–76 (2001)
20. Dawes, W.N.: Building Blocks Towards VR-Based Flow Sculpting. In: 43rd AIAA Aerospace Sciences Meeting & Exhibit, AIAA (January 2005)
21. Turk, G., O'Brien, J.: Modelling with Implicit Surfaces that Interpolate. ACM Transations on Graphics 21, 855–873 (2002)
22. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. Evolutionary Computation 6(2), 182–197 (2002)
23. Hirschen, K., Schäfer, M.: A Study on Evolutionary Multi-objective Optimization for Flow Geometry Design. Computational Mechanics 37(2), 131–141 (2006)
24. Ballester, P.J., Carter, J.N.: An Effective Real-Parameter Genetic Algorithm with Parent Centric Normal Crossover for Multimodal Optimisation. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 901–913. Springer, Heidelberg (2004)
25. Fogel, D.B.: Mutation Operators. In: Evolutionary Computation I: Basic Algorithms and Operators, pp. 237–255. Institute of Physics Publishing, Bristol, UK (2000)
26. Kellar, W.P.: Geometry Modeling in Computational Fluid Dynamics and Design Optimisation. PhD thesis, University of Cambridge (January 2003)