

ERCOFTAC

Bulletin

March 2015

102

European Research Community on Flow, Turbulence and Combustion

ERCOFTAC is a leading European association of research, education and industry groups in the technology of flow, turbulence and combustion. The main objectives of ERCOFTAC are: To promote joint efforts of European research institutes and industries with the aim of **exchanging technical and scientific information**; to promote **Pilot Centres** for collaboration, stimulation and application of

research across Europe; to stimulate, through the creation of **Special Interest Groups**, wellcoordinated European-wide research efforts on specific topics; to stimulate the creation of advanced training activities; and to be influential on funding agencies, governments, the European Commission and the European Parliament.

www.ercoftac.org

Honorary Presidents

Mathieu, J. Spalding, D.B.

Executive Committee

Chairman

Tomboulides, A.
University of Western
Macedonia, Greece
Tel: +30 246 105 6630
atompoulidis@uowm.gr
ananiast@auth.gr

First Deputy Chairman

Von Terzi, D.

Second Deputy Chairman

Hirsch, C.

Treasurer

Hämäläinen, J.

Deputy Treasurer

Ooms, G.

SPC Chairman

Geurts, B.J.

SPC Deputy Chairman

Sagaut, P.

KNC Chairman

Hutton, A. G.

KNC Deputy Chairman

Geuzaine, P.

Industrial Eng. Officer

Seoud, R.E.

Knowledge Base Editor

Rodi, W.

Bulletin Editor

Elsner, W.

ERCOFTAC Seat of the Organisation

Director

Hirsch, C.
Chaussée de la Hulpe 189
Terhulpesteenweg
B-1170 Brussels, Belgium
Tel: +32 2 643 3572
Fax: +32 2 647 9398
ado@ercoftac.be

Scientific Programme Committee

Chairman

Geurts, B.J.
University of Twente
Mathematical Sciences
PO Box 217
NL-7500 AE Enschede
The Netherlands
Tel: +31 53 489 4125
b.j.geurts@utwente.nl

Deputy Chairman

Sagaut, P.

Knowledge Network Committee

Chairman

Hutton, A. G.
Airbus Operations Ltd.
Building 07S
Filton, Bristol, BS99 7AR
United Kingdom
Tel: +44 117 936 7 590
anthony.hutton@airbus.com

Deputy Chairman

Geuzaine, P.

Industrial Eng. Officer

Seoud, R.E.
richard.seoud-ieo@ercoftac.org

ERCOFTAC Central Administration and Development Office (CADO)

Admin. Manager

Jakubczak, M.
PO Box 53877
London, SE27 7BR
United Kingdom
Tel: +44 203 602 8984
admin@cado-ercoftac.org
Skype: Ercoftacado

TABLE OF CONTENTS

Introduction to Special Theme: Design Optimization <i>K. C. Giannakoglou</i>	3
Optimization in Fluid Mechanics using Adjoint Methods and Evolutionary Algorithms <i>K.C. Giannakoglou, D.I. Papadimitriou, V.G. Asouti, X.S. Trompoukis, E.M. Papoutsis-Kiachagias, K.T. Tsiakas and I.S. Kavvadias</i>	4
An Aerodynamic Design Framework based on Algorithmic Differentiation <i>T. Albring, B.Y. Zhou, N.R. Gauger and M. Sagebaum</i>	10
Shape Optimization of U-bends for Internal Cooling Channels: an Overview <i>T. Verstraete</i>	17
Mission and Shape Optimization using modeFRONTIER: Application to Boomerang Throw <i>R. Russo, A. Clarich, C. Poloni and E. Nobile</i>	23
Adjoint-based Optimisation Methods for Automotive Applications <i>C. Othmer</i>	30
Multi-Objective Drag Reduction Design of Natural Laminar Airfoil with Shock Control Bump Shape using Evolutionary Algorithms and Pareto Games <i>Z. Tang, Y. Chen and J. Periaux</i>	34

EDITOR	Marek, M.
TECHNICAL EDITOR	Kuban, Ł.
CHAIRMAN	Elsner, W.
EDITORIAL BOARD	Armenio, V. Dick, E. Geurts, B.J.
DESIGN & LAYOUT	Borhani, N. Nichita, B.A.
COVER DESIGN	Aniszewski, W.

SUBMISSIONS

ERCOFTAC Bulletin
Institute of Thermal Machinery
Częstochowa University of Technology
Al. Armii Krajowej 21
42-201 Częstochowa
Poland
Tel: +48 343 250 507
Fax: +48 343 250 507
Email: ercoftac@imc.pcz.czyst.pl

HOSTED, PRINTED & DISTRIBUTED BY



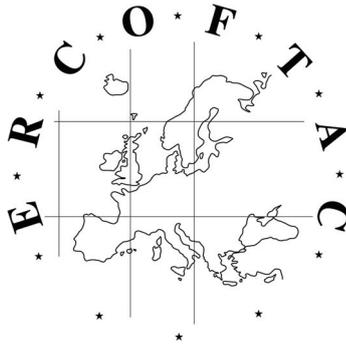
CZEŃSTOCHOWA UNIVERSITY OF TECHNOLOGY

The reader should note that the Editorial Board cannot accept responsibility for the accuracy of statements made by any contributing authors

NEXT ERCOFTAC EVENTS

ERCOFTAC Spring Festival
April 23rd 2015
Zürich, Switzerland

ERCOFTAC Committee Meetings
April 24th 2015
Zürich, Switzerland



The ERCOFTAC Best Practice Guidelines for Industrial Computational Fluid Dynamics

The Best Practice Guidelines (BPG) were commissioned by ERCOFTAC following an extensive consultation with European industry which revealed an urgent demand for such a document. The first edition was completed in January 2000 and constitutes generic advice on how to carry out quality CFD calculations. The BPG therefore address mesh design; construction of numerical boundary conditions where problem data is uncertain; mesh and model sensitivity checks; distinction between numerical and turbulence model inadequacy; preliminary information regarding the limitations of turbulence models etc. The aim is to encourage a common best practice by virtue of which separate analyses of the same problem, using the same model physics, should produce consistent results. Input and advice was sought from a wide cross-section of CFD specialists, eminent academics, end-users and, (particularly important) the leading commercial code vendors established in Europe. Thus, the final document can be considered to represent the consensus view of the European CFD community.

Inevitably, the Guidelines cannot cover every aspect of CFD in detail. They are intended to offer roughly those 20% of the most important general rules of advice that cover roughly 80% of the problems likely to be encountered. As such, they constitute essential information for the novice user and provide a basis for quality management and regulation of safety submissions which rely on CFD. Experience has also shown that they can often provide useful advice for the more experienced user. The technical content is limited to single-phase, compressible and incompressible, steady and unsteady, turbulent and laminar flow with and without heat transfer. Versions which are customised to other aspects of CFD (the remaining 20% of problems) are planned for the future.

The seven principle chapters of the document address numerical, convergence and round-off errors; turbulence modelling; application uncertainties; user errors; code errors; validation and sensitivity tests for CFD models and finally examples of the BPG applied in practice. In the first six of these, each of the different sources of error and uncertainty are examined and discussed, including references to important books, articles and reviews. Following the discussion sections, short simple bullet-point statements of advice are listed which provide clear guidance and are easily understandable without elaborate mathematics. As an illustrative example, an extract dealing with the use of turbulent wall functions is given below:

- Check that the correct form of the wall function is being used to take into account the wall roughness. An equivalent roughness height and a modified multiplier in the law of the wall must be used.
- Check the upper limit on y^+ . In the case of moderate Reynolds number, where the boundary layer only extends to y^+ of 300 to 500, there is no chance of accurately resolving the boundary layer if the first integration point is placed at a location with the value of y^+ of 100.

- Check the lower limit of y^+ . In the commonly used applications of wall functions, the meshing should be arranged so that the values of y^+ at all the wall-adjacent integration points is only slightly above the recommended lower limit given by the code developers, typically between 20 and 30 (the form usually assumed for the wall functions is not valid much below these values). This procedure offers the best chances to resolve the turbulent portion of the boundary layer. It should be noted that this criterion is impossible to satisfy close to separation or reattachment zones unless y^+ is based upon y^* .
- Exercise care when calculating the flow using different schemes or different codes with wall functions on the same mesh. Cell centred schemes have their integration points at different locations in a mesh cell than cell vertex schemes. Thus the y^+ value associated with a wall-adjacent cell differs according to which scheme is being used on the mesh.
- Check the resolution of the boundary layer. If boundary layer effects are important, it is recommended that the resolution of the boundary layer is checked after the computation. This can be achieved by a plot of the ratio between the turbulent to the molecular viscosity, which is high inside the boundary layer. Adequate boundary layer resolution requires at least 8-10 points in the layer.

All such statements of advice are gathered together at the end of the document to provide a 'Best Practice Checklist'. The examples chapter provides detailed expositions of eight test cases each one calculated by a code vendor (viz FLUENT, AEA Technology, Computational Dynamics, NUMECA) or code developer (viz Electricité de France, CEA, British Energy) and each of which highlights one or more specific points of advice arising in the BPG. These test cases range from natural convection in a cavity through to flow in a low speed centrifugal compressor and in an internal combustion engine valve.

Copies of the Best Practice Guidelines can be acquired from:

ERCOFTAC (CADO)
 PO Box 53877
 London, SE27 7BR
 United Kingdom
 Tel: +44 203 602 8984
 Email: magdalena.jakubczak@ercoftac.org

The price per copy (not including postage) is:

ERCOFTAC members		
First copy		<i>Free</i>
Subsequent copies		75 Euros
Students		75 Euros
Non-ERCOFTAC academics		
Non-ERCOFTAC industrial		140 Euros
EU/Non EU postage fee		230 Euros
		10/17 Euros

INTRODUCTION TO SPECIAL THEME: DESIGN OPTIMIZATION

K. C. Giannakoglou

National Technical University of Athens, Greece

kgianna@central.ntua.gr

Several years ago, in recognition of the growing importance of “Design Optimization” to industrial CFD/CSM applications, ERCOFTAC established a Special Interest Group, called SIG34. SIG34 focuses on the development and application of optimization methods, practically those used in CFD applications. Optimization is considered to be the means helping engineers in the search of optimal solutions to a given problem. One of the goals of SIG34 is to foster the spread of this technology in Europe. It forms a framework for closer co-operation between the research groups working on the subject and serves as a platform for the information exchange from science to industry. One of the most important goals is the transfer of information and, thereby, technology to companies not currently engaged in the technology. Among other, these goals are achieved through regular (biannual) workshops. SIG34 workshops are not limited to CFD applications, as one of the goals of the organisation is to benefit from the experience already gained in other areas such as structural mechanics and aeroacoustics. Main application areas are aeronautics, internal & external aerodynamics, (hydraulic & thermal) turbomachinery, car aerodynamics, combustion, structural mechanics, aeroacoustics, etc.

The first meeting of SIG34 took place in Nov. 1999, in Munich, followed by two training courses and an introductory course to design optimization (2000, 2001, 2003, respectively) in the same city. In March 2004, in Athens, it was the first time this course was upgraded to an International Conference (combined with an advanced course program). This type of Conferences or Introductory courses continued in the following years in Manchester (2005), Gran Canaria (2006), Trieste (2007), Berlin (2009), Manching (2011), Wolfsburg (2013).

The next course on Design Optimization will take place in Stockholm, in May 28-29, 2015 and will kindly be hosted by FOI. The course will provide (a) an overview of modern design optimization methods, (b) comprehensive discussions on the presented methods assisting industrial engineers to select the best-suited approach for solving their particular problems and (c) successfully treated examples in the areas of aeronautics, the automotive and turbomachinery industry. Lectures will be given by Prof. Dr. Kyriakos Giannakoglou (NTUA, Athens, Greece, Course Coordinator), Prof.

Dr. Nicolas R. Gauger, (TU Kaiserslautern, Germany), Prof. Dr. Tom Verstraete (VKI, Rhode Saint Genese, Belgium), Dr. Marcus Mayer (Rolls-Royce Deutschland, Germany), Dr. Olivier Amoignon (FOI, Stockholm, Sweden), Dr. Matteo Nicolich (ESTECO S.p.A, Italy), Prof. Anders Klarbring (Linköping University, Sweden), Prof. Martin Berggren (Umea University, Sweden) and Dr. Per Weinerfelt (SAAB Aeronautics, Sweden).

The present ERCOFTAC Bulletin includes six articles by six different research groups among the most active ones in SIG34. They are describing a variety of research activities related to the development modern optimization methods, either gradient-based or relying upon evolutionary algorithms, and their applications in various real-world problems. A few comments on the articles included in this volume follow. The article by K. Giannakoglou et al. reviews the development and use of CFD-based optimization methods including evolutionary algorithms and adjoint methods, with applications to thermal and hydraulic turbomachines, cars, high-lift devices, etc. The article by N. Gauger et al. presents an aerodynamic design framework in which algorithmic differentiation is applied to an open-source multi-physics solver to obtain design sensitivities, with results in both steady and unsteady aerodynamic optimization problems. The article by T. Verstraete is about the optimization of U bends for internal cooling channels inside turbine blades, addressing reduction in pressure losses and increase in heat transfer, based on evolutionary algorithms. The article by C. Poloni et al. illustrates a proof of concept of a multi-level optimization, involving parameters related to both geometry and mission control; in fact, an optimal boomerang is designed in terms of energy requested to throw the object, optimizing the shape of the boomerang and the throw parameters. The article by C. Othmer presents a number of different automotive optimization disciplines in Volkswagen, including topology and shape optimization of ducted flows (e. g. air ducts for cabin ventilation or engine ports), flow control and shape optimization in external aerodynamics and optimization of cooling and aeroacoustics. Finally, the article by J. Periaux et al. is on the multi-objective drag reduction design of natural laminar airfoil with shock control bump shape using evolutionary algorithms and Pareto games.

OPTIMIZATION IN FLUID MECHANICS USING ADJOINT METHODS AND EVOLUTIONARY ALGORITHMS

K.C. Giannakoglou, D.I. Papadimitriou, V.G. Asouti, X.S. Trompoukis,
E.M. Papoutsis-Kiachagias, K.T. Tsiakas and I.S. Kavvadias

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

Abstract

Research activities of the Parallel CFD & Optimization Unit (PCOpt) of the School of Mechanical Engineering of the National Technical University of Athens are mostly focusing on the development and use of CFD-based optimization methods; the parallelization and porting on GPUs of the developed methods is an active research area too. Evolutionary algorithms, supported by a variety of mechanisms to reduce their CPU cost, especially in large-scale problems with a great number of design variables, as well as deterministic (gradient-based) optimization methods, relying upon the adjoint method to compute the gradient of the objective and constraint functions, have been developed. These have been used in a variety of problems in turbomachines (thermal or hydraulic), aircraft and automotive applications as well as other problems in fluid mechanics. This article summarizes relevant research and achievements and demonstrates a few selected applications. The article is split into two parts: (a) activities and applications in the field of adjoint methods and (b) activities, methods and software related to evolutionary optimization. The hybridization of the two methods is possible and has been used in some cases.

1 Adjoint Methods

The development of adjoint methods at PCOpt is based on in-house flow solvers for both compressible and incompressible flows, using time-marching techniques for systems of hyperbolic equations and either the pseudo-compressibility technique or pressure-based methods for incompressible flows (the latter using OpenFOAM). On the in-house (U)RANS solver, both the continuous and discrete adjoint approaches have been developed; discrete adjoint is always based on hand-differentiation to minimize storage requirements. Direct differentiation and/or finite differences are implemented for the purpose of validation. The development of the adjoint to the OpenFOAM code is exclusively based on the continuous approach. The adjoint methods are dealing with different objective functions which are used to perform shape or topology optimization or to design flow control systems (via steady and unsteady jets); adjoint methods for use in robust design/optimization, where high-order derivatives must be computed, have also been developed. In constrained optimization problems, the adjoint method is used for both the objective and constraint functions.

The in-house primal and adjoint equation solvers are fully parallelized using the multi-domain approach and MPI. The same software has also been ported on Graph-

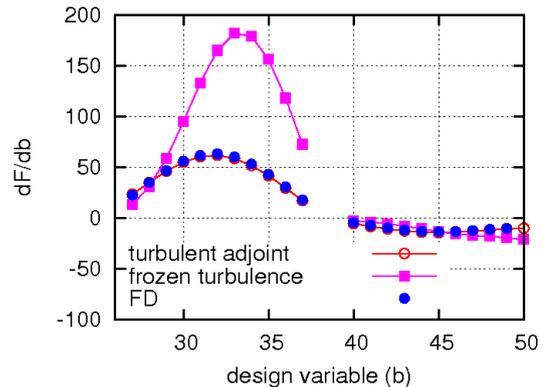


Figure 1: Shape optimization of a compressor cascade, $Re = 3.3 \times 10^5$, using the adjoint to the low- Re Spalart-Allmaras model. Sensitivity derivatives of the total pressure losses function F w.r.t. the coordinates b of the control points parameterizing the suction (first half) and pressure (second half of the horizontal axis) airfoil sides

ics Processing Units, using CUDA. The GPU-enabled software, for the solution of both the primal and the adjoint equations, runs about 60 times faster than the corresponding CPU solver on a single CPU core. The GPU solver employs Mixed Precision Arithmetics (MPA, [1, 2]) to minimize global memory transactions. Since, all residuals are computed with double precision arithmetics (DPA), MPA does not harm the solution accuracy. The high speed-up results from the use of single precision arithmetics (SPA) in storing the l.h.s. matrices.

This article focuses on continuous adjoint, unless stated otherwise. Objective functions are defined either along the controlled part of the domain boundaries (such as the walls of the body to be designed) or the inlet to/exit from the flow domain or, even, as field integrals.

In contrast to all previous works in continuous adjoint, PCOpt introduced the continuous adjoint method to both the mean-flow and turbulence equations. This has been done for several turbulence models (Spalart-Allmaras [3], $k-\epsilon$ [4] and $k-\omega$ SST [5] models) by overcoming the frequently made assumption of neglecting variations in turbulent viscosity. In many cases, the adjoint to the turbulence equations is really necessary. The omission of solving the adjoint to the turbulence model equation(s) may result to wrong and even wrongly signed sensitivities that may mislead the optimization algorithm.

The gain from overcoming the “frozen turbulence” assumption is shown in 1, concerning the flow within a compressor cascade, to be designed for minimum total pressure losses. On the starting geometry, the “frozen turbulence” assumption leads to wrong sensitivities.

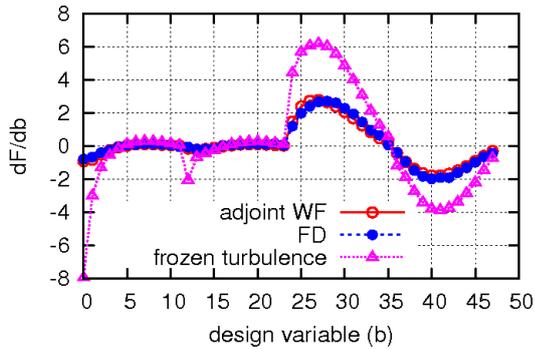


Figure 2: Adjoint to the Spalart–Allmaras model with WF; flow around the NACA0012 airfoil. The first 24 points correspond to the derivatives w.r.t. the x coordinates of the suction and pressure side control points while the last 24 to those w.r.t. the y coordinates

In industrial applications, the use of the wall function technique is a viable approach, used routinely in analysis and design projects. When the design is based on the adjoint method, considering the adjoint to the wall function model becomes necessary. The continuous adjoint method for the computation of sensitivity derivatives in aerodynamic optimization problems, modeled via turbulence models with wall functions, has been presented in [6], by introducing the adjoint wall functions. The proposed formulation led to the adjoint equations and boundary conditions expressed in terms of the adjoint friction velocity. The adjoint law of the wall bridges the gap between the solid wall and the first computational node off the wall during the solution of the adjoint PDEs.

The application of the adjoint approach to the high- Re Spalart–Allmaras model, [7], is shown in 2; it is about drag sensitivities of the isolated NACA0012 airfoil ($\alpha_\infty = 3^\circ$, $Re = 6 \times 10^6$). Sensitivities computed using the adjoint wall functions (WF) are compared to the outcome of finite differences (FD) and the adjoint method making the “frozen turbulence” assumption. The adjoint wall functions perfectly match the finite differences. Once reliable adjoint methods for the computation of the exact gradient were developed, next step was the development of methods to additionally compute the (exact) Hessian. There are two reasons for computing the Hessian: (a) to support the use of the (exact) Newton method within the optimization loop and (b) because this is needed in robust design, according to the method of moments. This was done using both discrete and continuous approaches, [8]. The use of direct differentiation (DD; for the gradient) followed by the adjoint method (for second derivatives; tangent–then–reverse in the terminology of Automatic Differentiation methods) computes the Hessian with the lowest cost.

This cost scales with the number of design variables. In small scale problems, the ability to compute the Hessian allows the use of the (exact) Newton method which, generally, outperforms steepest–descent or quasi-Newton methods. In cases with many design variables, however, since the CPU cost per cycle of the (exact) Newton method becomes prohibitively high an alternative scheme was proposed in [8]. According to this, the exact Hessian can be computed only once and, then, updated approximately through a quasi-Newton formula. This method usually outperforms (exact) Newton or quasi-Newton methods.

To avoid the computation of Hessian in large scale optimization problems, the truncated Newton algorithm

has been developed in [9]. Here, the Hessian computation is skipped and Hessian products with vectors are computed instead, since the Newton equation is solved via the conjugate gradient (CG) method. The adjoint approach followed by the DD of both the flow and adjoint equations is the most efficient way to compute them. Considering that the cost of solving either the adjoint or the DD equations is approximately equal to that of solving the flow equations, the cost per Newton iteration scales linearly with the number of CG steps, rather than the (much higher, in large scale problems) number of design variables.

The aforementioned work is related to optimization at given operating conditions and without geometrical or other imprecisions. To account for designs with acceptable performance in a stochastically changing environment, robust design methods have been developed. These rely on an appropriate cost function quantifying the way the objective function changes if the uncertain/environmental variables vary, based on an assumed probability density function. The second-order second-moment (SOSM) approach, requires the derivatives of the objective function w.r.t. the uncertain variables. Coupled with a gradient-based algorithm for the solution of robust design problems in aerodynamics, it has been developed in [10]. Since the cost function comprises first- and second-order sensitivity derivatives of the quantity of interest (lift, drag, etc.) w.r.t. the uncertain parameters, the use of a gradient-based method, which requires the sensitivities of the cost function w.r.t. the design variables, calls for the computation of third-order mixed derivatives. To compute them with the minimum CPU cost, a combination of DD and the adjoint method has been proposed.

In fluid mechanics, topology optimization is used for designing flow passages, connecting predefined inlets and outlets, with optimal performance based on selected criteria. The continuous adjoint approach for topology optimization in incompressible ducted flows with heat transfer has been developed. A variable porosity field, to be determined during the optimization, is the means to define the optimal topology. The developed methods account also for heat transfer and have taken all previous developments into account; for instance, they may handle turbulent flows, by fully differentiating the turbulence model. [11] was the first publication on topology optimization using continuous adjoint methods and exact gradients in turbulent flows. Since in various problems, such as those dealing with the design of manifolds, constraints on the outlet flow direction or mass flow rates and mean outlet temperatures must be imposed, the adjoint method has been extended to these constraint functions as well. The continuous adjoint approach has the advantage of providing closed form expressions for the sensitivity derivatives with good interpretation of the various terms. For instance, working with the viscous losses, the dominant sensitivity term is the inner product of the primal and adjoint velocities. Such a remark sheds light to the mechanism contributing to the elimination or reduction of recirculation areas which are associated with losses. The angle formed between the primal and adjoint velocity vectors determines the sign of the sensitivity derivative computed at each grid node. In the absence of adjoint backflow, areas with negative primal streamwise velocities yield negative inner products and sensitivity values which tend to increase the corresponding local porosity values. Hence, the recirculation area is solidified and, progressively, eliminated.

Adjoint methods have also been employed for the optimization of flow control configurations. The role of flow control, based on suction/blowing/synthetic jets, is to prevent or delay separation, control transition to turbulent flow, suppress or enhance turbulence, control shock waves, etc. The sensitivities of the total pressure losses function with respect to the flow control parameters, i.e. the suction/blowing velocities, on a fixed geometry, have been computed in [12], using continuous adjoint. Based on the computed gradients, one may choose the optimal position and jet type. High sensitivity values indicate the optimal positions of jets whereas their signs determine whether suction or blowing must be used, [12].

In 3 to 6, some industrial applications are presented. The first case deals with the blade optimization of a 3D peripheral compressor cascade for min entropy losses within the flow passage, [8]. The second one demonstrates the shape optimization of a Francis turbine runner in order to suppress cavitation, [13], the third the shape optimization of a Volkswagen car for min drag force, [14], and the last one the topology optimization of an air-conditioning duct targeting min total pressure losses.

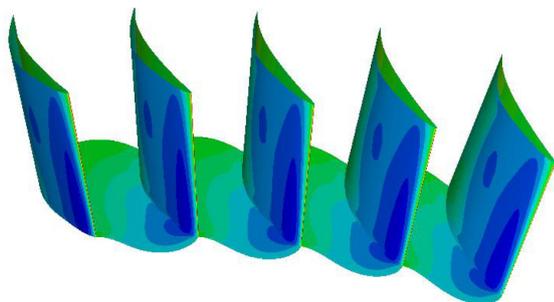


Figure 3: Shape optimization of a 3D peripheral compressor cascade, targeting min entropy generation rate within the flow passage. The coordinates of the 3D NURBS control points parameterizing the shape of the compressor blade are the design variables. Pressure distribution along the compressor blade corresponding to the optimized shape computed using the continuous adjoint approach

2 Stochastic Optimization Methods

In the field of stochastic population-based optimization methods, PCOpt mostly works on the cost reduction of Evolutionary Algorithms (EAs). EAs, in their standard form, require a great number of evaluations (here based on expensive CFD runs) to reach the optimal solution(s). In real-world applications, this increases the optimization turnaround time a lot. Techniques to reduce the CPU cost of an EA-based optimization are really necessary so as to make them a viable tool in an industrial environment. The relevant research includes the use of surrogate evaluation models (or metamodels), distributed and/or hierarchical algorithms and parallel and/or asynchronous variants of EAs. They are all incorporated in the general purpose optimization platform EASY (Evolutionary Algorithm SYstem, <http://velos0.ltt.mech.ntua.gr/EASY>) developed by PCOpt.

In Metamodel-Assisted EAs (MAEAs), “local metamodels” (radial basis function networks, multilayer per-

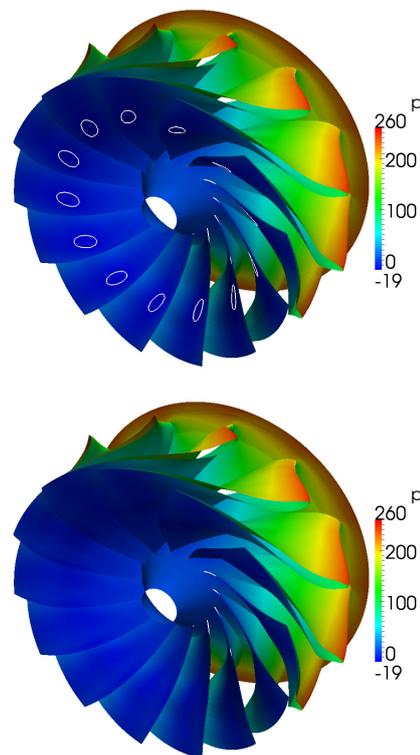


Figure 4: Optimization of a Francis runner blade targeting cavitation suppression. The objective function involves min/max operations which makes it non-differentiable and requires special treatment. Top: pressure distribution over the initial blading; white isolines encircle the cavitated areas. Bottom: pressure distribution over the optimized blading, where the cavitated area has practically disappeared. From [13]

ceptrons, kriging etc.), on-line trained for each and every new individual generated during the evolution, are used according to the Inexact Pre-Evaluation (IPE) framework, [15, 16, 17]. For all but the first few generations, which are exclusively based on the exact evaluation model, every new population member is pre-evaluated on the low-cost metamodel, trained on the fly on neighboring already evaluated candidate solutions. Based on the outcome of these pre-evaluations, the most promising population members are identified to undergo evaluation on the CFD s/w, before proceeding to the next generation. In MAEAs, the selection of training patterns per candidate solution is important since it affects the prediction ability of metamodels. This selection is based on self-organizing maps (SOMs) and an iterative combination of unsupervised and supervised learning, [17]. The gain offered by the use of metamodels in the EA-based design of a four-element airfoil is demonstrated in 7. For the same CPU cost, MAEAs outperform EAs.

In engineering optimization problems with a great number of design variables, both EAs and MAEAs suffer from the curse of dimensionality. A possible remedy to this problem is the implementation of principal component analysis (PCA) on the current elite set in each generation. The PCA provides useful information to (a) better guide the application of the evolution operators and (b) reduce the number of sensory units of artificial neural networks used as metamodels (in MAEAs), [18, 19].

By using the PCA to guide the evolution operators (EA(PCA)) the design space becomes temporarily

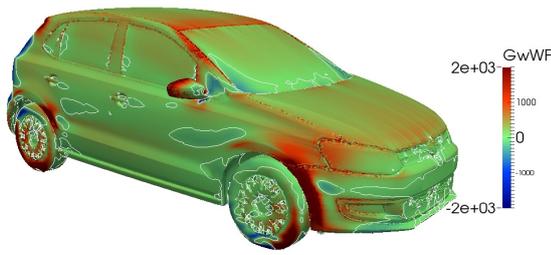


Figure 5: Sensitivity map of the VW Polo car targeting min drag force. Red areas should be displaced inwards in order to reduce drag whereas blue areas outwards. Green areas have an almost zero sensitivity magnitude and have, thus, a very small optimization potential. White isolines indicate sensitivity sign changes

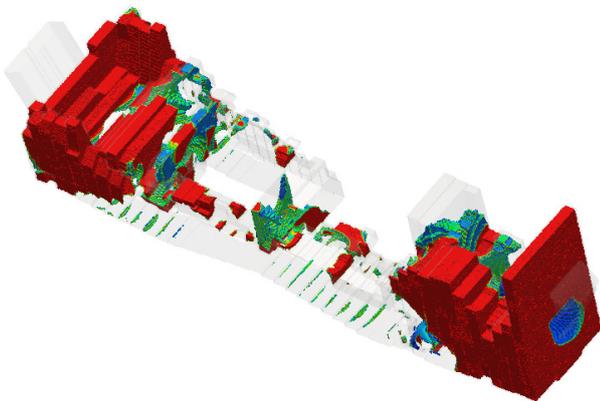


Figure 6: Topology optimization of an air-conditioning duct of a passenger car targeting min total pressure losses. The flow enters the domain from the left and exits from the right. White (transparent) areas have been left intact. Red areas correspond to high porosity values and should be solidified in order to avoid high energy losses. Intermediate colours indicate lower porosity values

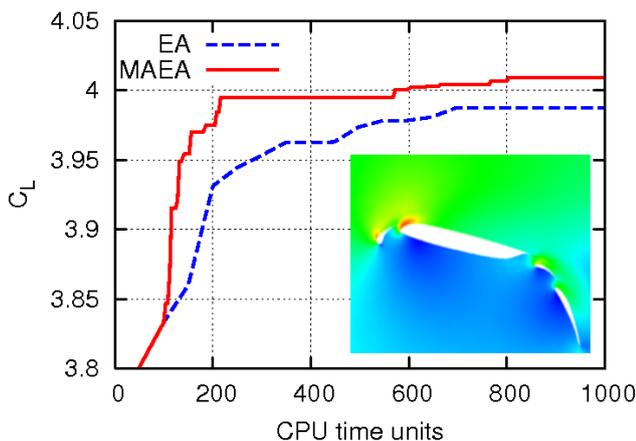


Figure 7: Optimal deployment of a four-element airfoil for max lift coefficient. The design variables are the displacements and rotations of the two flaps and the slat (9 in total). Mach field for the optimal configuration

aligned with the principal directions and the crossover and mutation operators are applied on the so “rotated” individuals. This alignment leads to an optimization

problem with an “almost separable” objective function, which is highly beneficial.

In MAEAs, during the metamodel training, the PCA is used to reduce the problem dimension. The variances of the design variables are used to identify the directions along which the elite members are less or more scattered; high variances indicate scattered data and vice-versa. Based on this, the metamodel sensory units corresponding to the directions of the design space with high variances are filtered out. This reduction in the number of input parameters increases the prediction accuracy of metamodels and accelerates the training process. The training patterns are rotated and, then, their components associated with the higher eigenvalues are excluded from the training. This method is referred to as M(PCA)AEA. If the PCA is used to assist both the application of the evolution operators and the metamodel training, this is referred to as M(PCA)AEA(PCA). An example of the use of PCA for the design of a Francis runner with 372 design variables is shown in 8. MAEA(PCA) performs better than the MAEA whereas M(PCA)AEA(PCA) outperforms any other variant.

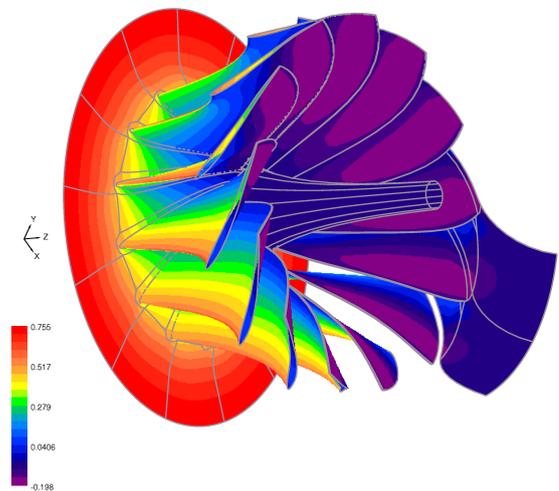
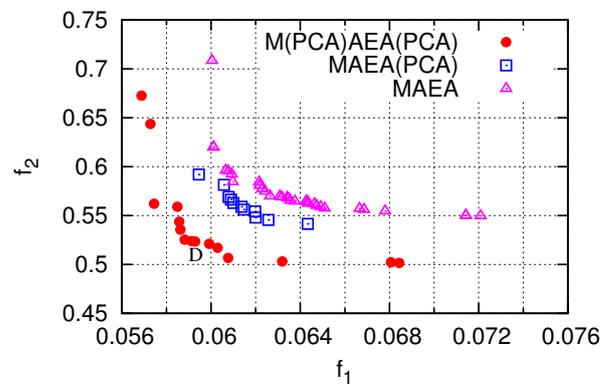


Figure 8: Two-objective design of a Francis runner at three operating points for optimizing the outlet velocity profile (min f_1) and the blade loading (min f_2). Top: Comparison of the fronts of non-dominated solutions computed by the MAEA, MAEA(PCA) and M(PCA)AEA(PCA), at the same CPU cost. Bottom: Pressure field over the Francis runner, at the best efficiency operating point for the non-dominated solution D. From [19]

Another method for reducing the optimization turnaround time is via distributed search, i.e. the so-called Distributed EAs (DEAs). A DEA handles a

few sub-populations or demes evolving in semi-isolation. The demes communicate according to the directed graph topology and exchange their best performing individuals according to a user-defined inter-deme migration policy. Distributed schemes outperform EAs and their coupling with MAEAs (DMAEAs) leads to an even better performance.

Hierarchical optimization methods rely on the regular exchange of information between successive levels (usually two, occasionally three). Each level can be associated with different (in terms of accuracy/computational cost) evaluation software, or problem parameterization and/or search tool. Hierarchical or multilevel EAs (HEAs) offer additional CPU cost reduction and can be used in combination with MAEAs (HMAEAs) or DMAEAs (HDMAEAs), [20, 21, 22, 23].

In “multilevel evaluation” mode, an HEA uses different evaluation software on each level. The lower level(s) are responsible for detecting near-optimal solutions in at low CPU cost before delivering them to the higher level(s) for further refinement. Inexpensive, low-fidelity evaluation models are associated with lower levels while, on the higher levels, evaluation models of higher fidelity and CPU cost are employed. An application of a multilevel evaluation HEA is presented in 9. The distributed HEA (DHEA) is shown to perform better than the conventional EA, while the additional use of metamodels, i.e. DHMAEA, outperforms both EA and DHEA. In the “multilevel search” mode, each level of the HEA handles a different search technique. Stochastic search is often used on the lower levels for the exploration of the design space, leaving the refinement of promising solutions to gradient-based methods on the higher levels. Finally, in “multilevel parameterization” mode, each level of the HEA is associated with a different set of design variables. On the lowest level, a few design variables are considered. On the next levels, the problem dimension increases. The detailed problem parameterization is used on the highest level only.

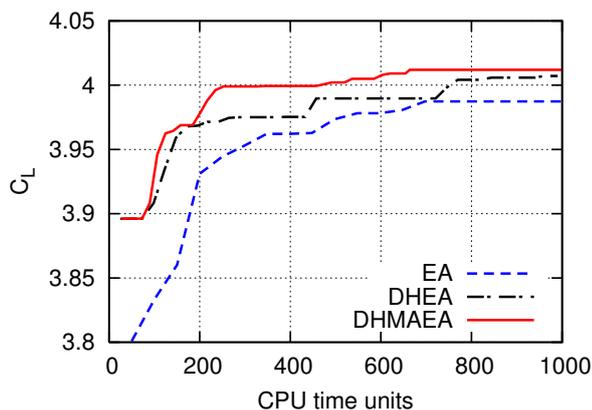


Figure 9: Optimal deployment of a four-element airfoil for max lift coefficient. Comparison of the convergence history of EA, DHEA and DHMAEA (“multilevel evaluation”) in terms of CPU cost. The evaluation tool is the in-house GPU-enabled RANS solver. The low level uses its SPA variant while the high level its MPA variant. The CPU cost ratio of the two variants is about 1:2 on the same computational grid

In all the aforementioned variants of EAs, the concurrent evaluation of candidate solutions in the same generation on different processors, [24], reduces further the optimization turnaround time. However, it does not maximize parallel efficiency due to the synchronization

barrier at the end of each generation. To fully exploit all the available computational resources by overcoming the “generation” barrier, asynchronous EAs (AEAs) have been developed, [25]. The efficiency of AEAs is improved by employing metamodels (AMAEA), according to an appropriately adapted IPE technique, [26]. A representative example is shown in 10.

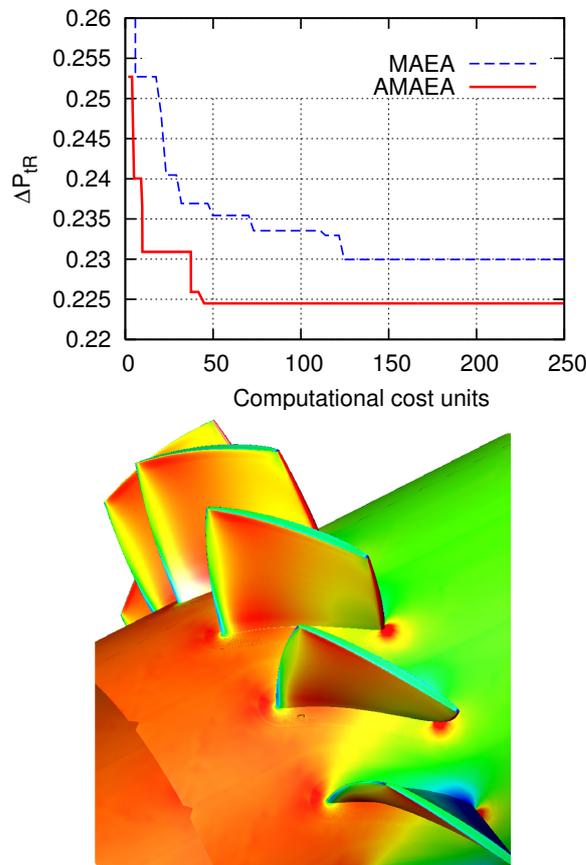


Figure 10: Optimization of a peripheral compressor cascade for minimum viscous losses. Top: Convergence histories of MAEA and AMAEA. Bottom: Pressure distribution on the optimal geometry

References

- [1] I. Kampolis, X. Trompoukis, V. Asouti, and K. Giannakoglou, “CFD-based analysis and two-level aerodynamic optimization on graphics processing units,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 9-12, pp. 712–722, 2010.
- [2] X. Trompoukis, V. Asouti, I. Kampolis, and K. Giannakoglou, *CUDA implementation of Vertex-Centered, Finite Volume CFD methods on Unstructured Grids with Flow Control Applications*, ch. 17. Morgan Kaufmann, 2011.
- [3] A. Zymaris, D. Papadimitriou, K. Giannakoglou, and C. Othmer, “Continuous adjoint approach to the Spalart-Allmaras turbulence model for incompressible flows,” *Computers & Fluids*, vol. 38, no. 8, pp. 1528–1538, 2009.
- [4] E. Papoutsis-Kiachagias, A. Zymaris, I. Kavvadias, D. Papadimitriou, and K. Giannakoglou, “The continuous adjoint approach to the $k-\epsilon$ turbulence

- model for shape optimization and optimal active control of turbulent flows,” *Engineering Optimization, to appear*, 2014.
- [5] I. Kavvadias, E. Papoutsis-Kiachagias, G. Dimitrakopoulos, and K. Giannakoglou, “The continuous adjoint approach to the $k-\omega$ sst turbulence model with applications in shape optimization,” *Engineering Optimization, to appear*, 2014.
 - [6] A. Zymaris, D. Papadimitriou, K. Giannakoglou, and C. Othmer, “Adjoint wall functions: A new concept for use in aerodynamic shape optimization,” *Journal of Computational Physics*, vol. 229, no. 13, pp. 5228–5245, 2010.
 - [7] E. Papoutsis-Kiachagias and K. Giannakoglou, “Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications,” *Archives of Computational Methods in Engineering, to appear*, 2014.
 - [8] D. Papadimitriou and K. Giannakoglou, “Aerodynamic shape optimization using first and second order adjoint and direct approaches,” *Archives of Computational Methods in Engineering*, vol. 15, no. 4, pp. 447–488, 2008.
 - [9] D. Papadimitriou and K. Giannakoglou, “Aerodynamic design using the truncated newton algorithm and the continuous adjoint approach,” *International Journal for Numerical Methods in Fluids*, vol. 68, no. 6, pp. 724–739, 2012.
 - [10] D. Papadimitriou and K. Giannakoglou, “Third-order sensitivity analysis for robust aerodynamic design using continuous adjoint,” *International Journal for Numerical Methods in Fluids*, vol. 71, no. 5, pp. 652–670, 2013.
 - [11] E. Kontoleonos, E. Papoutsis-Kiachagias, A. Zymaris, D. Papadimitriou, and K. Giannakoglou, “Adjoint-based constrained topology optimization for viscous flows, including heat transfer,” *Engineering Optimization*, vol. 45, no. 8, pp. 941–961, 2013.
 - [12] A. Zymaris, D. Papadimitriou, E. Papoutsis-Kiachagias, K. Giannakoglou, and C. Othmer, “The continuous adjoint method as a guide for the design of flow control systems based on jets,” *Engineering Computations*, vol. 40, no. 4, pp. 494–520, 2013.
 - [13] E. Papoutsis-Kiachagias, S. Kyriacou, and K. Giannakoglou, “The continuous adjoint method for the design of hydraulic turbomachines,” *Computer Methods in Applied Mechanics and Engineering*, vol. 278, pp. 612–639, 2014.
 - [14] E. Papoutsis-Kiachagias, K. Giannakoglou, and C. Othmer, “Adjoint wall functions - validation and application to vehicle aerodynamics,” in *11th World Congress on Computational Mechanics*, (Barcelona, Spain), July 21-25 2014.
 - [15] K. Giannakoglou, A. Giotis, and M. Karakasis, “Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters,” *Inverse Problems in Engineering*, vol. 9, no. 4, pp. 389–412, 2001.
 - [16] M. Karakasis, A. Giotis, and K. Giannakoglou, “Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization,” *International Journal for Numerical Methods in Fluids*, vol. 43, no. 10-11, pp. 1149–1166, 2003.
 - [17] M. Karakasis and K. Giannakoglou, “On the use of metamodel-assisted, multi-objective evolutionary algorithms,” *Engineering Optimization*, vol. 38, no. 8, pp. 941–957, 2006.
 - [18] S. Kyriacou, S. Weissenberger, and K. Giannakoglou, “Design of a matrix hydraulic turbine using a metamodel-assisted evolutionary algorithm with PCA-driven evolution operators,” *International Journal of Mathematical Modelling and Numerical Optimization*, vol. 3, no. 2, pp. 45–63, 2012.
 - [19] S. Kyriacou, V. Asouti, and K. Giannakoglou, “Efficient PCA-driven EAs and metamodel-assisted EAs, with applications in turbomachinery,” *Engineering Optimization*, vol. 46, no. 7, pp. 895–911, 2014.
 - [20] M. Karakasis, D. Koubogiannis, and K. Giannakoglou, “Hierarchical distributed evolutionary algorithms in shape optimization,” *International Journal for Numerical Methods in Fluids*, vol. 53, no. 3, pp. 455–469, 2007.
 - [21] I. Kampolis and K. Giannakoglou, “A multilevel approach to single- and multi-objective aerodynamic optimization,” *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 33-40, pp. 2963–2975, 2008.
 - [22] I. Kampolis and K. Giannakoglou, “Distributed evolutionary algorithms with hierarchical evaluation,” *Engineering Optimization*, vol. 41, no. 11, pp. 1037–1049, 2009.
 - [23] I. Kampolis and K. Giannakoglou, “Synergetic use of different evaluation, parameterization and search tools within a multilevel optimization platform,” *Applied Soft Computing*, vol. 11, no. 1, pp. 645–651, 2011.
 - [24] P. Liakopoulos, I. Kampolis, and K. Giannakoglou, “Grid-enabled, hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization,” *Future Generation Computer Systems*, vol. 24, no. 7, pp. 701–708, 2008.
 - [25] V. Asouti and K. Giannakoglou, “Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes,” *Engineering Optimization*, vol. 41, no. 3, pp. 241–257, 2009.
 - [26] V. Asouti, I. Kampolis, and K. Giannakoglou, “A grid-enabled asynchronous metamodel-assisted evolutionary algorithm for aerodynamic optimization,” *Genetic Programming and Evolvable Machines*, vol. 10, no. 3, pp. 373–389, 2009.

AN AERODYNAMIC DESIGN FRAMEWORK BASED ON ALGORITHMIC DIFFERENTIATION

T. Albring, B.Y. Zhou, N.R. Gauger and M. Sagebaum

TU Kaiserslautern, Chair for Scientific Computing, 67663 Kaiserslautern
{tim.albring,yuxiang.zhou,nicolas.gauger,max.sagebaum}@scicomp.uni-kl.de

Abstract

Efficient and robust optimization methods are essential for the design of future aircraft. In this work, we develop an aerodynamic design framework in which algorithmic differentiation is applied to the open-source multi-physics solver SU² to obtain design sensitivities. We obtain a consistent discrete adjoint solver which directly inherits the convergence properties of the primal flow solver due to the differentiation of the entire non-linear fixed-point iterator. Results from both steady and unsteady aerodynamic optimization problems are presented.

1 Introduction

During the last few decades efficient aerodynamic optimization methods have become increasingly important. Here, adjoint methods are quite successful as they require only one evaluation of the underlying adjoint system to obtain the sensitivities for an arbitrary number of design variables [1, 2]. Although the continuous version, based on the continuous integral formulation of the flow equations, is efficient in terms of memory consumption and run-time, it is difficult to extend the approach to handle additional equations like turbulence or transition models, for example. Furthermore, there is a slight inconsistency between the calculated gradient and the objective function because the method yields a discrete approximation of the gradient of the analytic objective function, rather than of the discretized objective function. As a result the optimization is likely to fail to converge further once in the vicinity of a local minimum. Nevertheless, this approach is used frequently nowadays for shape optimization problems. A possible remedy for the latter problem is the use of discrete adjoint methods [3, 4] where the optimal control theory is applied to the discretized flow equations. Still, it suffers from the same drawback, namely the difficult extension to complex flow models if the traditional approach based on the transposed of the state Jacobian is deployed and it may also lead to consistency problems if the numerical methods are not properly linearized.

In this work we will consider a discrete approach based on the principle of Algorithmic Differentiation (AD) [5]. Although there were several approaches in the past that applied AD in the field of aerodynamics [6, 7, 8, 9], most of them suffered either from poor performance (using AD tools based on the Operator overloading approach) or from limited flexibility (source-code transformation). To overcome these issues we employ modern C++ features to automatically generate a representation of the computational graph of each statement at compile-time.

The aforementioned properties allow essentially the application of this approach to any available flow solver

as long as it is written in C or C++. However, it will prove its strength particularly in an constantly evolving multi-disciplinary framework. Therefore we applied this approach to the SU² open-source code that comprises a complete self-contained optimization framework for aerodynamic design [10, 11]. Initially developed at Stanford University it now exhibits collaborations from all over the world and receives regular updates and improvements. Originally based on the continuous adjoint method, a lot of the already present input/output routines, data structures and optimization methods could be reused for the discrete adjoint solver and the forward mode differentiation mode, such that it integrates now smoothly into the existing framework.

In this work we will present some of the features of this framework along with an introduction of the used methods.

2 The Aerodynamic Design Chain

In the following we will give a short overview of the optimization problem and the already implemented methods. Note that the discussions below are made on the basis of steady problems. The unsteady counterpart can be derived analogously and is omitted here for the sake of brevity. 1 shows a simplified representation of the aerodynamic design chain. The components of the design vector α can for example be chosen as the amplitudes of Hicks-Henne functions [12] in 2D or as the control points of the Free-Form deformation method [13] for 3D problems. According to a movement of the surface based on the current values of the design variables, a mesh deformation routine using the Linear Elasticity method creates a new mesh X . The flow solver then evaluates the flow field U and the objective function J . Using this environment, the optimization problem for steady problems can be stated as

$$\min_{\alpha} J(U(\alpha), X(\alpha)) \quad (1)$$

$$\text{subject to } R(U(\alpha), X(\alpha)) = 0 \quad (2)$$

where $R(U)$ is the flow residual arising from the semi-discretization of the governing equation.

The discretization of the compressible RANS equations and the turbulence models is performed using the Finite-Volume method on a vertex-based median-dual grid. Several numerical fluxes such as JST, Roe, AUSM etc. are implemented and flux and slope limiters enable second-order space integration. By using an implicit Euler scheme for the time integration, followed by a linearization about the known time level we obtain the

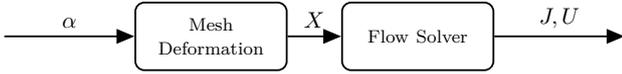


Figure 1: Simplified representation of the aerodynamic design chain

following linear system to be solved in each iteration n :

$$\left(D^n + \frac{\partial R(U^n)}{\partial U^n} \right) \Delta U^n = -R(U^n). \quad (3)$$

Here, $R(U^n)$ is the residual resulting from the space integration, $\Delta U_i^n := U_i^{n+1} - U_i^n$ and

$$(D^n)_{ij} := \frac{|\Omega_i|}{\Delta t_i^n} \delta_{ij}, \quad \left(\frac{\partial R(U^n)}{\partial U^n} \right)_{ij} := \frac{\partial R_i(U^n)}{\partial U_j^n}. \quad (4)$$

Ω_i represents the volume of the cell i and Δt_i^n is the (pseudo) time-step that may be different in each cell due to the local time-stepping technique. If non-linear multi-grid acceleration is used, then equation (3) is additionally discretized and solved on consecutively coarser meshes in each iteration to find a correction to be applied to U^{n+1} .

3 Differentiation of the Design Chain

The application of AD to the aerodynamic design chain described in section 2 is similar to the Operator Overloading approach [9] as it essentially requires the replacement of all variables in the routines involved in the computation of the objective function by a datatype that not only stores the numerical value, but also enables the breakdown of each statement into elementary operations, such as summation, multiplication, sine/cosine etc. For the reverse mode, the stored sequence of operations and expressions can then be traversed in reverse order to compute the gradient. For the forward mode the derivatives are computed along the evaluation of an expression. Therefore, it does not require as much additional memory as the reverse mode. For a detailed introduction we refer to the relevant literature, for example to the book of Griewank and Walther [5]. Template metaprogramming features of C++ are then applied to implement the Curiously Recurring Template Pattern [14] that allows to build the computational graph for each statement at compile-time. For that purpose we use a modified version of the Adept C++ library [15]. Additionally, the current implementation is fully capable of handling the MPI communication for the use in parallel computations, both, for reverse and forward mode.

One particular advantage of the operator overloading approach is the possibility to develop an easy maintainable adjoint solver. In particular, we gain a high level of flexibility that allows for the automatic adaption of the adjoint solver to modifications and extensions of the flow solver. Even the incorporation of turbulence or transition models can be done with ease or grid movement strategies like moving walls can be included as long as they are implemented and are ready to be used in the flow solver.

4 The Discrete Adjoint Solver for Steady Flows

An important fact is that if D^n would be zero ($\Delta t_i^n \rightarrow \infty$), then (3) would be identical to a step of Newton's method to solve the non-linear system $R(U) = 0$. However, using the implicit Euler discretization we naturally end up with a damped Newton method for solving $R(U) = 0$. Hence, if convergence is achieved, the resulting solution U^* only depends on the right-hand side of (3), that is, the residual $R(U)$. The left-hand side can therefore be any reasonable approximation to the flow Jacobian $\frac{\partial R}{\partial U}$. This can be made clear by transforming the flow equation (3) into a fixed point equation $U = G(U)$, such that feasible solutions can be computed from the iteration

$$U^{n+1} = G(U^n) := U^n - P(U^n)R(U^n) \quad (5)$$

with the preconditioner P defined as

$$P(U) := \left(D + \frac{\partial \tilde{R}(U)}{\partial U} \right)^{-1}. \quad (6)$$

Note, for simplicity the multi-grid method is neglected in this representation of P and G . The tilde indicates that this might be an approximation to the exact Jacobian. It is naturally to assume that G is stationary only at feasible points, i.e.

$$R(U^*) = 0 \Leftrightarrow U^* = G(U^*). \quad (7)$$

By the Banach fixed-point theorem, recurrence (5) converges, if G is contractive, i.e. if $\|\frac{\partial G}{\partial U}\| < 1$ in a suitable matrix norm. In advanced CFD codes, like SU², there are several approximations often seen to reduce the complexity:

- Use of first order approximations of the implicit terms, even though a higher order spatial discretization is applied on the right-hand side.
- Consistent linearized treatment of the boundary conditions is typically neglected.
- Only approximate solutions of the linear system (3) are obtained.

If traditional discrete adjoint methods are used [16], however, these approximations are not valid anymore, since they result in a linear system involving the exact flow Jacobian $\frac{\partial R}{\partial U}$ to be solved for the adjoint variables. To circumvent this problem, Korivi et al. [17] proposed a method for solving the adjoint system that resembles the iterative flow solver and permits the use of the same approximative Jacobian. For the derivation of the proposed discrete adjoint solver, we adopt this approach and combine it with the efficient evaluation of the occurring gradients using AD.

Since the computational mesh is subject to change, we consider now all functions additionally depended on X . To formally handle the surface and mesh deformation, we add it as a constraint to the original optimization problem (1) - (2) by using the equation $M(\alpha) = X$. A similar way of dealing with the mesh sensitivities was originally proposed by Nielsen and Park [18]. However, in the present case we do not make any assumptions on the structure of M , except that is differentiable. Then

the optimization problem finally takes the form

$$\min_{\alpha} J(U(\alpha), X(\alpha)) \quad (8)$$

$$\text{subject to } U(\alpha) = G(U(\alpha), X(\alpha)) \quad (9)$$

$$X(\alpha) = M(\alpha). \quad (10)$$

We can define the Lagrangian associated to this problem as

$$L(\alpha, U, X, \bar{U}, \bar{X}) = J(U, X) + (G(U, X) - U)^T \bar{U} + (M(\alpha) - X)^T \bar{X} \quad (11)$$

$$= N(U, \bar{U}, X) - U^T \bar{U} + (M(\alpha) - X)^T \bar{X} \quad (12)$$

where N is the shifted Lagrangian

$$N(U, \bar{U}, X) := J(U, X) + G^T(U, X)\bar{U}. \quad (13)$$

If we differentiate L with respect to α using the chain rule, we can choose the adjoint variables \bar{X} and \bar{U} in such a way, that the terms $\frac{\partial U}{\partial \alpha}$ and $\frac{\partial X}{\partial \alpha}$ can be eliminated. This leads to the following equations for \bar{U} and \bar{X} :

$$\bar{U} = \frac{\partial}{\partial U} N(U, \bar{U}, X) \quad (14)$$

$$= \frac{\partial}{\partial U} J(U, X) + \frac{\partial}{\partial U} G^T(U, X)\bar{U}$$

$$\bar{X} = \frac{\partial}{\partial X} N(U, \bar{U}, X) \quad (15)$$

$$= \frac{\partial}{\partial X} J(U, X) + \frac{\partial}{\partial X} G^T(U, X)\bar{U}$$

Finally, the derivative of the Lagrangian, that is, the total derivative of J , reduces to

$$\frac{dL}{d\alpha} = \frac{dJ}{d\alpha} = \frac{d}{d\alpha} M^T(\alpha)\bar{X}. \quad (16)$$

Equation (14) is a fixed-point equation in \bar{U} and can be solved in the style of the flow solver using the iteration

$$\bar{U}^{n+1} = \frac{\partial}{\partial U} N(U^*, \bar{U}^n, X) \quad (17)$$

once we have found a numerical solution $U = U^*$ of equation (5). Since G is a contractive function if the flow solver has reached a certain level of convergence (i.e. $\|\frac{\partial G}{\partial U}\| < 1$ in some suitable matrix norm), also $\frac{\partial N}{\partial U}$ will be contractive since

$$\left\| \frac{\partial}{\partial \bar{U}} \left(\frac{\partial N^T}{\partial U} \right) \right\| = \left\| \frac{\partial G^T}{\partial U} \right\| = \left\| \frac{\partial G}{\partial U} \right\| < 1. \quad (18)$$

Thus, it directly inherits the convergence properties of the flow solver. The sensitivity equation (15) can efficiently be evaluated using AD along with the computation of \bar{U} as they require the derivatives of the same functions. An important fact is, that since N in (14) is evaluated at a fixed U^* , it is only necessary to store the computational graph once at the first iteration. In subsequent iterations we can simply re-evaluate the gradient computation routine provided by the AD tool. Similarly, the mesh sensitivity equation (16) requires the application of AD to the mesh deformation routines.

4.1 Simplified Recurrence and the Linear Solver

Since the efficiency of the approach directly depends on the number of active variables, that is, variables that are involved in the computation of L and with dependency of X or U , considering some parts of the code as passive can greatly reduce the memory requirements and the runtime of the adjoint solver. For the derivative of G with respect to U holds

$$\frac{\partial}{\partial U} G(U, X) = I - P(U, X) \frac{\partial}{\partial U} R(U, X) - \frac{\partial}{\partial U} P(U, X) R(U, X). \quad (19)$$

If $\frac{\partial P}{\partial U}$ is uniformly bounded, which is the case if P is continuously differentiable [5], the last term will disappear as $R(U, X)$ converges to zero. Hence, we may drop the last term and use the *simplified recurrence*

$$\bar{U}^{n+1} = \frac{\partial}{\partial U} J(U^*, X) + \left(I - P(U^*, X) \frac{\partial}{\partial U} \tilde{R}(U^*, X) \right)^T \bar{U}^n. \quad (20)$$

The same holds for the derivative with respect to X and the evaluation of \bar{X} . This explicit separation of the preconditioner P from G is only possible if no multi-grid method is used. Still, it can easily be shown that this approach is also possible if G is more complex. See for example the book of Griewank [5]. It should be noted, that this simplification introduces a small error in the adjoint variables and therefore in the resulting gradient if the flow residual is not sufficiently reduced. However, we noticed that it is negligible even in case of engineering precision (for a flow residual in the order of 10^{-4} the relative error was smaller than 0.05%).

The simplified recurrence can easily be implemented in the context of AD if one uses an analytic differentiation of the linear solution process, i.e. the application of P on R in (5). For a detailed explanation we refer the interested reader to the paper of Giles [19]. Instead of recording all expression during the execution of the linear solver, which should always be avoided, this essentially will result in solving a linear system during the reverse pass for computing $\frac{\partial N}{\partial U}$ involving the transposed of the preconditioner P . By omitting the update of the gradient values of the entries of P after solving this linear system we directly end up with the simplified recurrence (20).

4.2 Comment on Performance

An important role for the application in productive environments plays of course the efficiency in terms of runtime and memory requirements. Without going to much into detail, currently the factor to get the gradient compared to one flow solution is about 1.2 – 3.0, depending on the numerical methods used in the flow solver. Especially the use of the multi-grid method pushes this factor in direction of the upper bound. However, future research will investigate the explicit handling of this kind of convergence acceleration methods to further increase the efficiency of the adjoint solver.

5 Results

In this section we present some optimizations that incorporate the AD-based gradient computations. For problems involving steady flows, the discrete adjoint solver

presented in section 4 is applied. At the moment, checkpointing has also been implemented for the reverse mode at each time iteration. However, to alleviate the still-large memory overhead, checkpointing or the simplified recurrence from Section 4.1 needs to be implemented within each inner iteration of the dual-time stepping scheme. Since this implementation requires still some work we have to fall back to the forward mode of AD to compute the necessary gradients for unsteady flows. Similar to the extension of the steady flow solver to an unsteady flow solver, it is possible to reuse most of the already implemented methods for the unsteady adjoint solver. For this reason the authors are quite confident to release this feature within the very near future.

The already implemented design framework that is built around SU^2 , which is based on Python and SciPy, is extended to support the differentiated routines, such that the application of the AD-based optimization framework is not more difficult than the already existent one. In particular, the optimizer used in the following sections is still based on the Han-Powell quasi-Newton method [20] implemented in the SciPy Python library.

5.1 Transonic Turbulent Flow

The first test case uses the supercritical RAE2822 airfoil as the baseline design in two-dimensional transonic turbulent flow conditions for drag reduction. The Reynolds number is 6.5 million based on the unit chord length, the farfield Mach number is $Ma_\infty = 0.729$ and airfoil is inclined at an angle of attack of 2.31° . The computational domain is discretized using a hybrid mesh consisting of a structured part around the airfoil and an unstructured part in the mid and farfield with a total of 59,976 elements and $y^+ \approx 1$. The first grid point of the airfoil surface is at a distance of $1E-5$ chords, and the far field boundary is located approximately one hundred chord lengths away from the airfoil. Characteristic-based far-field boundary conditions are enforced on the far-field boundary, and a no-slip, adiabatic boundary condition is enforced on the airfoil. The convective fluxes are computed using the Jameson-Schmidt-Turkel (JST) scheme while the turbulent viscosity is calculated using the SST $k-\omega$ model. Convergence to steady state is accelerated using a three level multi-grid W-cycle.

For the optimization the amplitudes of 38 equidistantly distributed Hicks-Henne bump functions are used to discretize the design space. The lift is constrained to remain constant at $C_{l_{base}} = 0.6901$, while the pitching moment about the quarter-chord point is forced to stay below the baseline value of $C_{mz_{base}} = 0.0856$. Furthermore, a geometric constraint is applied to keep the area of the airfoil constant during the optimization.

The RAE2822 airfoil is known to have a strong shock appearing on the suction side of the airfoil in this flow regime, that is indicated by the discontinuity in the computed pressure distribution C_p shown in 2 at the mid-chord position. Since it is well-known, that a local minimum of the drag coefficient is found by eliminating this shock it often serves as a reference test case.

To validate the discrete adjoint solver presented in Section 4 we performed a comparison of the drag gradients obtained using this solver and by finite differences (step size of 10^{-5}) that is shown in 3 for the baseline design. Despite a very slight disagreement around the shock position due to the inaccuracy of the finite difference approximation, we get an excellent agreement, indicating a consistent gradient computation. In 4 the drag, lift and pitching moment values during the optimization are

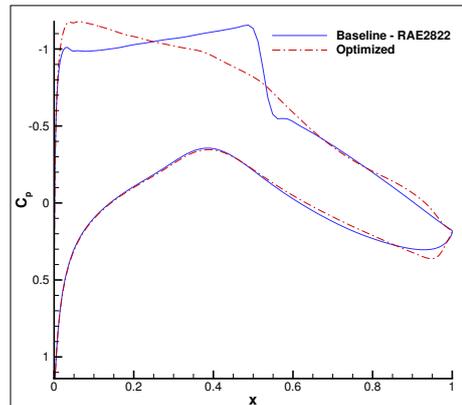


Figure 2: Pressure distribution C_p for the baseline RAE2822 airfoil and the optimized design

plotted. After 19 design cycles the drag is reduced from $C_{d_{base}} = 0.0129$ to $C_{d_{opt}} = 0.0112$, which translates to a relative reduction of 12%, while the lift is perfectly maintained. Interestingly, up to the 11th design cycle the pitching moment is pushing against its constraint value and suddenly drops down in line with a further drop in the drag value. This may indicate that we have found a better local optimum with the pitching moment not being at the boundary of the design space. If we consider the optimized pressure distribution in 2, we notice, as expected, a smooth distribution over the whole surface with increased generation of lift at the leading edge, that directly translates to the decreased pitching moment.

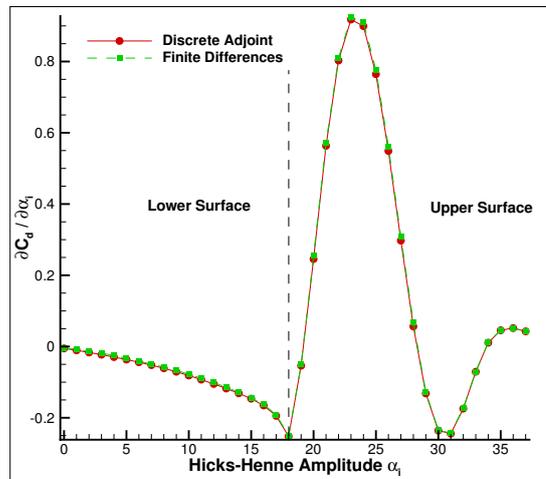


Figure 3: Comparison of drag gradient obtained by the discrete adjoint method and finite differences

5.2 Inviscid 3D Flow

The second test case features the drag reduction of the ONERA M6 airfoil in transonic, inviscid flow. The free-stream Mach number is set to $Ma_\infty = 0.8395$ with an angle of attack of 3.06° . The computational domain is discretized using an unstructured grid with 582,752 elements, with the surface consisting of 19,894 triangles. In order to model an extension, the airfoil is attached to a symmetry boundary. The convective flux is again discretized using the JST scheme.

The baseline airfoil has a strong shock structure appearing on the upper surface of the airfoil that extends from the leading edge and is visible in the contour plot of the pressure distribution in 5(a).

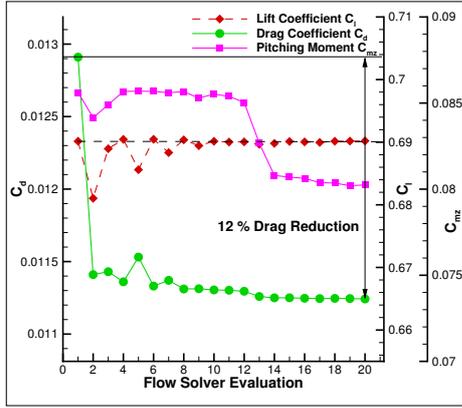
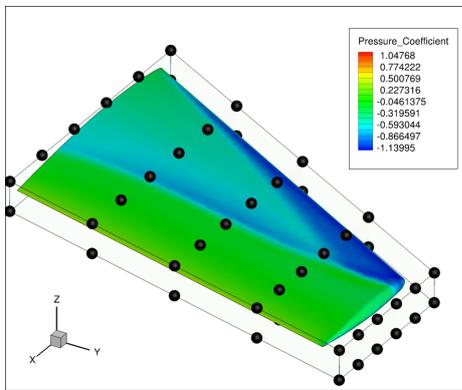
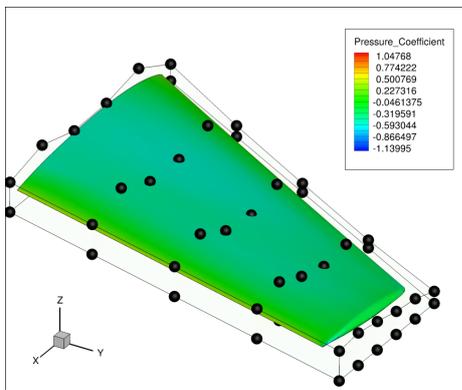


Figure 4: Drag, lift and moment values during optimization

The FFD control points are distributed as shown in 5(a). Note, for the optimization only a subset of the control points on the upper surface is used and only a movement in the z -direction is allowed. To be more precise, the last row of control points near the trailing edge is fixed, resulting in a total of 25 design variables. The lift is constrained to remain larger than or equal to the baseline value of $C_{L_{base}} = 0.2864$. 6 shows the development of the drag and lift coefficient during the optimization. After 11 design cycles the process has reached a state where significant improvements are not noticeable anymore resulting in a reduction of 35% compared to the initial value. The lift has not changed and remained essentially at the constraint initial value. The optimization successfully removed the shock, leaving a continuous pressure distribution over the surface (see 5(b)).



(a) Baseline Design



(b) Optimized Design

Figure 5: Pressure distribution C_p and position of the FFD control points for the inviscid test case

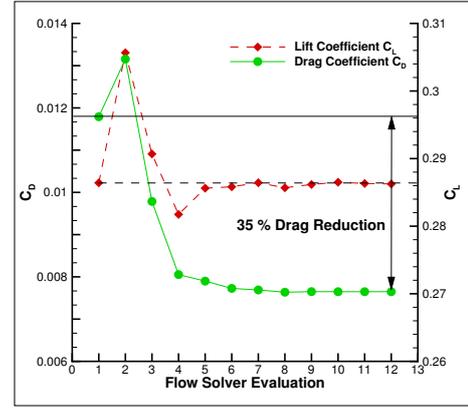


Figure 6: Optimization history for the inviscid test case

5.3 Pitching Airfoil in Transonic Turbulent Flow Condition

In this test case, we consider again the drag minimization problem of an RAE2822 airfoil in transonic ($Ma_\infty = 0.8$) and turbulent ($Re = 6.5 \times 10^6$) flow regime, but this time pitching about its quarter-chord point with a mean angle of attack of 2.0 degrees, an amplitude of 3.0 degrees and a frequency of 251.3 rad/s. The mesh used is an unstructured, O-grid that wraps around the RAE 2822 airfoil. It has 22,842 elements in total with 192 edges making up the airfoil boundary and 40 edges along the far field boundary. A time step of 0.001 is used for the dual time-stepping, resulting in 25 time steps per period of oscillation for a total of 8 periods. All other numerical settings coincide with the test case in section 5.1.

i	$\frac{\partial \bar{C}_D}{\partial \alpha_i}$ (AD)	$\frac{\partial \bar{C}_D}{\partial \alpha_i}$ (FD)	$\frac{\partial \bar{C}_L}{\partial \alpha_i}$ (AD)	$\frac{\partial \bar{C}_L}{\partial \alpha_i}$ (FD)
1	0.1744	0.1746	-4.7621	-4.7646
2	0.0751	0.0748	-5.7623	-5.7872
3	-0.2833	-0.2839	-3.6506	-3.6711
4	1.0985	1.1002	-4.8842	-4.9178
5	0.2652	0.2705	-11.718	-11.728
6	-0.3166	-0.3226	0.8805	0.8081

Table 1: Comparison between the gradients obtained by forward mode of AD and finite differences

A total of 6 Hicks-Henne bump functions are chosen as design variables and they are equally spaced along the upper and lower surfaces of the airfoil (3 on each of the upper and lower surfaces). The objective function is defined as the time-averaged drag \bar{C}_d within the periodic steady state, as denoted by the optimization window on 8. The averaged lift \bar{C}_l is constrained to remain above 0.4. Additionally, a thickness constraint is imposed to ensure that the maximum thickness of the airfoil remains constant. On 1, lift and drag sensitivities computed using the forward-mode AD are validated against those computed using finite difference.

The optimization is performed over 11 design cycles. The final airfoil shape achieves a 29% reduction in time-averaged drag, while the time-averaged lift constraint is fulfilled, as shown on 7. A comparison in the time-histories of the lift and drag performances between the baseline RAE2822 and the optimized airfoil profiles is shown on 8. The effect of the optimization can be visualized by comparing the Mach number contour of the airfoils at their mean (9 (b) and 10 (b)) and maximum (9 (c) and 10 (c)) angles of attack – the shock strength

on the upper surface has been significantly reduced, although it is apparent that the presence of the lift and thickness constraints as well as the low number of design variables do not allow sufficient degrees of freedom for the optimizer to fully remove the shocks. Note that at all three angles shown on (9 and 10), the shock position has moved further downstream in the optimized airfoil, reducing the extent of shock-induced boundary layer separation on the airfoil surface. Consequently, the new airfoil achieves lower drag within the optimization window as shown on 8.

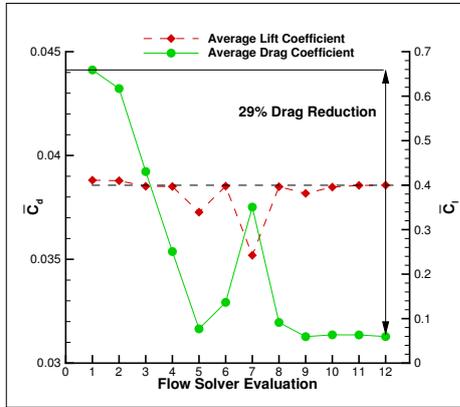


Figure 7: Optimization history of the pitching RAE2822 airfoil in transonic turbulent flow

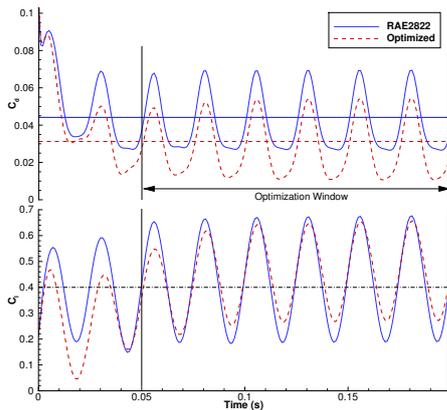
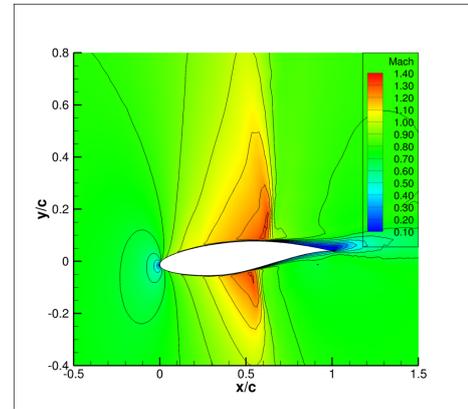


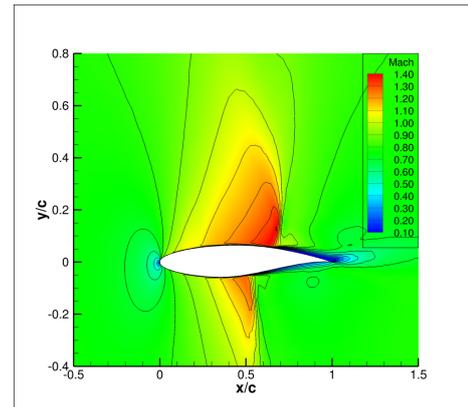
Figure 8: Time history of C_d and C_l for the baseline airfoil and the optimized airfoil. Horizontal lines indicate \bar{C}_d and \bar{C}_l

6 Summary & Outlook

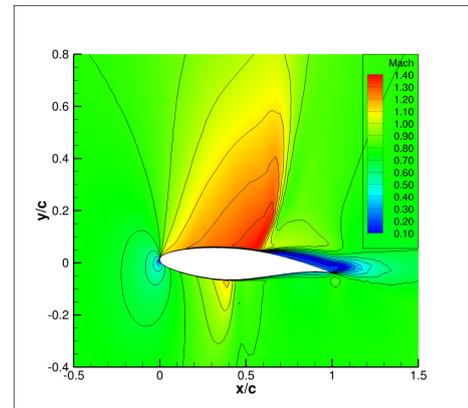
In this paper, we demonstrate an aerodynamic design framework in which algorithmic differentiation is applied to the open-source multi-physics solver SU² to obtain design sensitivities. We obtain a consistent discrete adjoint solver which directly inherits the convergence properties of the primal flow solver due to the differentiation of the entire nonlinear fixed-point iterator. This includes the differentiation of various turbulence models. Two validation and application cases are presented for steady problems – a 2-D viscous flow with a SST $k-\omega$ turbulence model over the RAE2822 airfoil and a 3-D inviscid flow over the ONERA M6 wing. As a first step in the extension to an unsteady adjoint solver, we apply the forward-mode AD to the optimization of a transonic pitching airfoil. This will serve as a reference test case for reverse-mode AD in the next step.



(a) $\alpha = -1^\circ$



(b) $\alpha = 2^\circ$



(c) $\alpha = 5^\circ$

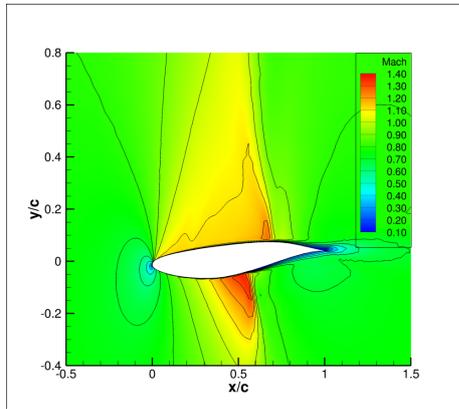
Figure 9: Mach number contour of the baseline RAE2822 airfoil at various time instances over one period

Acknowledgment

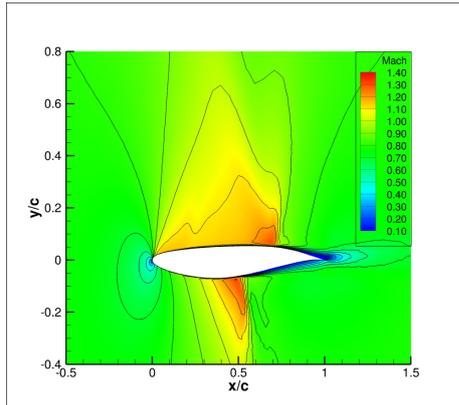
The authors would like to gratefully acknowledge the support from Thomas D. Economou, Francisco Palacios and Juan J. Alonso of the SU² development team at Stanford University.

References

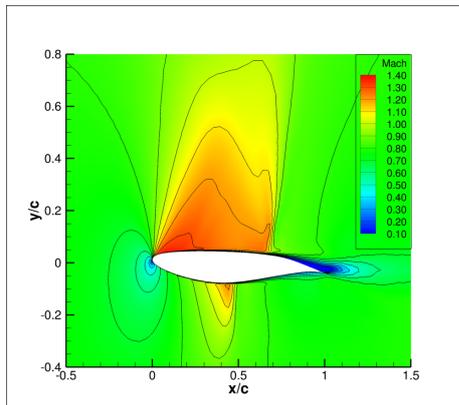
- [1] A. Jameson, “Aerodynamic design via control theory,” *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.
- [2] J. Reuther and A. Jameson, “Control theory based airfoil design using the Euler equations.” AIAA Paper 94-4272, 1994.



(a) $\alpha = -1^\circ$



(b) $\alpha = 2^\circ$



(c) $\alpha = 5^\circ$

Figure 10: Mach number contour of the optimized airfoil cycle at various time instances over one period

[3] E. J. Nielsen and W. K. Anderson, "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equation." AIAA Paper 98-4809, 1998.

[4] J. Elliott and J. Peraire, "Aerodynamic Design Using Unstructured Meshes." AIAA Paper 96-1941, 1996.

[5] A. Griewank and A. Walther, *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, second ed., 2008.

[6] B. Mohammadi, J.-M. Malé, and N. Rostaing-Schmidt, "Automatic differentiation in direct and reverse modes: Application to optimum shapes design in fluid mechanics," in *Computational Dif-*

ferentiation: Techniques, Applications, and Tools, pp. 309–318, Philadelphia, Penn.: SIAM, 1996.

[7] B. Mohammadi, "Optimal shape design, reverse mode of automatic differentiation and turbulence." AIAA Paper 97-0099, 1997.

[8] P. Hovland, B. Mohammadi, and C. Bischof, "Automatic Differentiation and Navier-Stokes Computations," *Computation Methods for Optimal Design and Control*, pp. 265–284, 1998.

[9] N. R. Gauger, A. Walther, C. Moldenhauer, and M. Widhalm, "Automatic differentiation of an entire design chain for aerodynamic shape optimization," *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 96, pp. 454–461, 2008.

[10] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. Taylor, and J. J. Alonso, "Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design," *AIAA Paper*, vol. 287, p. 2013, 2013.

[11] F. Palacios, T. Economon, A. Aranake, S. R. Copeland, A. K. Lonkar, and J. A. et al, "Stanford university unstructured (su2): Open-source analysis and design technology for turbulent flows," *AIAA Paper 2014-0243*, vol. 287, p. 2014, 2014.

[12] R. M. Hicks and P. A. Henne, "Wing design by numerical optimization," *Journal of Aircraft*, vol. 15, pp. 407–412, 1978.

[13] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," *SIGGRAPH Comput. Graph.*, vol. 20, pp. 151–160, Aug. 1986.

[14] J. O. Coplien, "Curiously recurring template patterns," *C++ Rep.*, vol. 7, pp. 24–27, Feb. 1995.

[15] R. J. Hogan, "Fast reverse-mode automatic differentiation using expression templates in C++," *ACM Transactions on Mathematical Software*, vol. 40, pp. 26:1–26:24, jun 2014.

[16] M. B. Giles and N. A. Pierce, "An introduction to the adjoint approach to design," *Flow, Turbulence and Combustion*, vol. 65, pp. 393–415, 2000.

[17] V. M. Korivi and A. C. Taylor, "An incremental strategy for calculating consistent discrete cfd sensitivity derivatives," 1992.

[18] E. J. Nielsen and M. A. Park, "Using an adjoint approach to eliminate mesh sensitivities in computational design," in *Proceedings of 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 2005. AIAA Paper 2005-0491.

[19] M. Giles, "Collected matrix derivative results for forward and reverse mode algorithmic differentiation," in *Advances in Automatic Differentiation*, vol. 64, pp. 35–44, Springer Berlin Heidelberg, 2008.

[20] D. Kraft, *A software package for sequential quadratic programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht, Wiss. Berichtswesen d. DFVLR, 1988.

SHAPE OPTIMIZATION OF U-BENDS FOR INTERNAL COOLING CHANNELS: AN OVERVIEW

T. Verstraete

Von Karman Institute
tom.verstraete@vki.ac.be

Abstract

U-bends are found in various ducted applications in which the flow direction needs to be turned 180 degrees. The present work looks into the application of these U-bends for internal cooling channels inside turbine blades, where two major aspects need to be carefully addressed: pressure loss needs to be reduced while heat transfer needs to be enhanced.

An overview of different shape optimization studies is given with the aim to improve the performance of the standard U-bend consisting of two circular arcs. Different optimization methodologies were used in this study ranging from single-objective Evolutionary Algorithms (EA), with or without acceleration by surrogate model, to multi-objective EAs, to finally gradient based adjoint optimization. The difference in computational cost is compared for the different applications combined with their advantages and disadvantages, and finally one optimal configuration is experimentally verified and compared to the numerical predicted improvement.

1 Introduction

Turbine blades are since long equipped with internal cooling channels as an effective way to increase cycle efficiency of gas turbines by augmenting the firing temperature. These cooling channels are in a vast majority of cases implemented through a serpentine scheme, in which one single ducted flow passes multiple times the blade span. Near the extremities of the blade span the flow inside the duct is turned 180 degrees through U-bends. The coolant air is bled from the high pressure compressor which bypasses the combustor and enters the turbine blade through its root.

Since the coolant air needs to be pressurized while it does not participate in the work extraction in the turbine, it represents a loss in global cycle efficiency. As a result, the internal cooling channels need to simultaneously allow for a high heat transfer at the lowest possible pressure loss. The U-bends present in the serpentine cooling channels are responsible for up to 25% of the total pressure loss in the channel and merit a profound attention, as witnessed by numerous experimental studies [1, 2, 3, 4].

This paper presents an overview of several numerical optimization studies performed, including:

- Single-objective optimization with EA (2D)
- Single-objective optimization with EA accelerated by a surrogate model (3D)

- Multi-objective optimization by EA accelerated by a surrogate model (3D)
- adjoint based optimization (2D)

2 U-bend test Case

2.1 Geometry

The U-bend under investigation is typical of internal cooling channels. The baseline geometry is shown in 1. It consists of a circular U-bend with radius ratio of 0.76, a hydraulic diameter of 0.075 meter and an aspect ratio of 1. The Reynolds number is 40.000 and the Mach number of 0.05 allows using an incompressible assumption. The shape of the inner and outer curve is allowed to be changed but needs to remain inside the bounding box shown in the figure, which restricts the height and width of possible changes to account for structural limits. The distance between both cooling channels is not subject to optimization, as well as the hydraulic diameter.

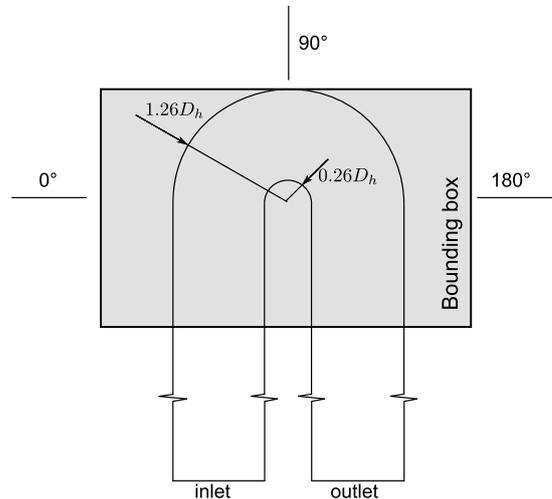


Figure 1: Baseline geometry, definition of the area in which the shape is allowed to change

2.2 Parameterisation

The U-bend has been parameterised by Bézier curves for which the movements of the control points are the design variables. For the adjoint based optimization, both the inner and outer curve of the U-bend have been represented by two separate continuous curves containing 20 control points. This results in a total of 80 design variables (x and y coordinate of each of the 2 curves), which

is unfeasible for the EA based optimization. Therefore, the EA based optimization uses a segmented approach, for which a total of 4 Bézier curves represent the inner or outer curve. Each Bézier curve only comprises of 4 control points, while relations between the control points are imposed to assure a sufficient degree of continuity between the curves. 2 shows the parameterisation of the outer curve.

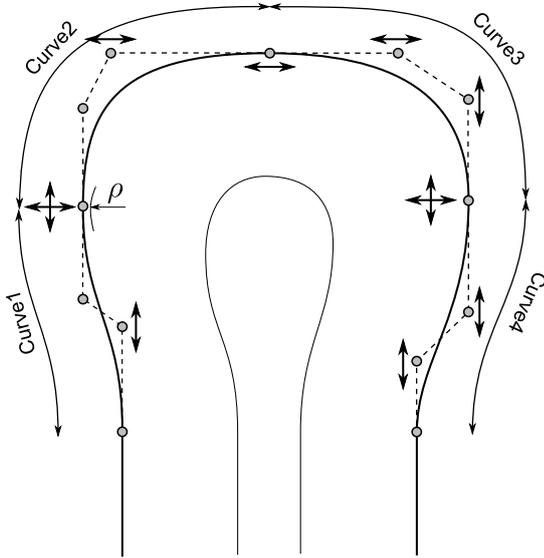


Figure 2: Parameterization of the outer curve

2.3 Performance Evaluation

The simpleFoam solver from OpenFoam [5] is used to evaluate the incompressible Navier-Stokes equations. The Launder-Sharma low-Reynolds $k-\epsilon$ turbulence model is used. The $k-\epsilon$ model “is arguably the simplest complete turbulence model” (Pope [6]), is implemented in most commercial software and is one of the most broadly employed at industrial level. Its performance is reasonably satisfactory in shear flows with small effects of streamwise pressure gradients and streamline curvatures, but far from these assumptions, it can fail badly. However it has been selected for the present application due to its large diffusion: given that the proposed methodology is apt for industrial problems, it was the intention to demonstrate its potential in conditions that are representative of real-life design practice.

At the inlet a fully developed velocity profile is imposed, together with values of k and ϵ for the turbulence model. Both are computed based on a turbulence intensity of 5% measured in the lab. At the outlet the static pressure is imposed.

The U-bend optimization is driven by the minimization of the pressure drop introduced by the U-bend and in case of the multi-objective optimization the maximization of the heat transfer is additionally considered. Both objective functions are defined as:

$$\text{Min } Obj_1 = \frac{P_{total}^{inlet} - P_{total}^{outlet}}{\frac{1}{2}\rho \cdot v_{ref}^2} \quad (1)$$

$$\text{Max } Obj_2 = \frac{Q}{Q_{ref}} \quad (2)$$

where Q is the total heat transferred to the fluid and P_{total} is the total pressure which is computed as the mass flow averaged quantity at the inlet respectively outlet of

the domain, positioned 8 hydraulic diameters away from the U-bend.

3 Optimization Strategies

Two distinct optimization strategies have been used to the applied test case and allow comparison between the different techniques. On the one hand, Evolutionary Algorithms (EA's) are used as a non-deterministic optimization method. These methods benefit a wide community of users and are relatively easy to understand and implement, factors which have contributed to the large diffusion of the method. On the other hand, a deterministic gradient based optimization has been used, in which the gradient is computed efficiently through the adjoint method.

3.1 Evolutionary Algorithms

Evolutionary Algorithms (EA) have been developed in the late sixties by J. Holland [7] and I. Rechenberg [8]. They are inspired from Darwinian evolution, whereby populations of individuals evolve over a search space and adapt to the environment by the use of different mechanisms such as mutation, crossover and selection. Individuals with a higher fitness have more chance to survive and/or get reproduced.

This natural process is translated to engineering problems in several steps. First, the shape is parameterised (as discussed in section 2.2) which defines an analogy to the DNA of an individual. This ensures that a unique combination of design parameters will represent a unique shape. Next, the operations that enable EA's to generate offspring such as mutation and crossover need to be translated. There exists a wide variety of techniques for this, which give rise to various classes of EA methods. Genetic Algorithms (GAs) for instance usually allow two individuals from a parent generation to reproduce two children through a crossover process on the design variables with analogy from nature. In Differential Evolution (DE) on the contrary, as many as four individuals are required to produce one child per parent, here analogy with nature is lost. In a final step, a selection procedure needs to be introduced, imposing a pressure on the population in which fitter designs have more chance to be selected for reproduction, while non-fit designs have larger probability to become extinct and disappear in the next generation. Potentially, additional mechanisms can be introduced to increase convergence through keeping a healthy diversity among the individuals of the population and by making sure that good individuals do not get lost accidentally by bad luck. Eventually, these algorithms can be easily modified to deal with multi-objective optimization problems, which identify the Pareto front.

EA methods are capable to work with noisy objective functions and can find global optima of multimodal problems. They however require a large number of function evaluations, which leads to unacceptable large computational costs in case the objective function depends on CFD evaluations. Especially for large design spaces the computational cost can be prohibitive, restricting the use of these methods to only a small design space. Typically, up to 20 or slightly more design variables can be considered, depending on the level of interaction between the different parameters.

To reduce the computational cost, very often a surrogate model is introduced, which is a sort of interpolation tool using the already analyzed individuals by CFD. The

surrogate model performs the same task as the high fidelity CFD analysis, but at a very low computational cost. However, it is less accurate, especially for an evaluation far away from the already analyzed points in the design space.

The implementation of the surrogate into the optimization system depends on how the system deals with the inaccuracy of the model. The technique used in the present work uses the surrogate model as an evaluation tool during the entire evolutionary process. After several generations the evolution is stopped and the best individual is analyzed by the expensive analysis tool. This technique is referred to as the “offline trained surrogate model”. The difference between the predicted value of the surrogate model and the high fidelity tool is a direct measure for the accuracy of the surrogate model. Usually at the start this difference is rather large. The newly evaluated individual is added to the database used for the interpolation and the surrogate model will be more accurate in the region where previously the EA was predicting a minimum. This feedback is the most essential part of the algorithm as it makes the system self-learning. It mimics the human designer which learns from his mistakes on previous designs.

3.2 Gradient Based Optimization

Optimization methods that use gradient information are iterative methods that continuously alter the shape with small perturbations. The basic idea behind these methods is that through the knowledge of the gradient the direction can be found in which the design variables need to be changed in order to obtain an improved design. Small modifications to the design variables are required, as the gradient will only provide a linear approximation to the real objective function and remains only valid in the neighborhood of the current design. The simplest gradient based optimization method is the steepest descent, which modifies the design variables in the direction of the steepest descent, given by the opposite direction of the gradient. Although it has the lowest convergence rate of all gradient based methods, it is still an attractive method and will also be used in this work

The most complex part of this type of method is however to compute the gradient information, especially for problems which require the solution of partial differential equations to compute the objective. This can be achieved through a forward method, such as for instance finite differences, complex variable perturbation or algorithmic differentiation. In brief, these methods perturb one by one each design variable and compute the difference with the unperturbed design. The main drawback is that the computational cost is proportional to the number of design variables, requiring n additional CFD computations for n design variables.

The computational cost can however be dramatically reduced by reverse or adjoint methods, which require a cost proportional to only one CFD computation to obtain the gradient information, irrespective the number of design variables. In the case of continuous adjoint methods, a new set of linear partial differential equations needs to be solved after convergence of the CFD analysis, as for instance derived in [9]. Then the gradient can be computed with small effort. It is evident that such methods are preferred, as they allow for an efficient computation of the gradient even for extremely large design spaces (literally every grid point on the boundary of the shape can become a design variable). They however require a large development and implementation cost,

which has been one of the major reasons for their reduced usage compared to EA or other gradient free methods. Additional disadvantages of gradient based methods is the local search, which allows only to find the nearby local optimum in case of multimodal problems.

4 Results

4.1 Single-objective EA

A single objective Differential Evolution (DE) algorithm is applied to the U-Bend optimization. Since DE requires a large number of evaluations when not supported by a surrogate model, the problem is viewed in 2D to reduce the cost per CFD computation. In 3 the convergence history of the optimization can be seen. A total of 100 populations of 40 individuals each need to be performed in order to obtain convergence. This means a total of 4000 CFD computations. A reduction 35% in total pressure loss could be achieved.

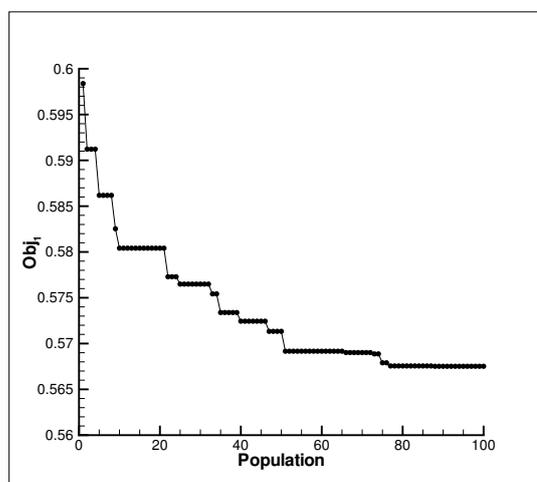


Figure 3: Convergence of the EA without surrogate model

4.2 Single-objective EA Assisted by Surrogate Model

A reduction in the computational cost can be obtained by using a surrogate model. In 4 the convergence history of a surrogate model assisted DE optimization is shown. It compares the surrogate model prediction (here a Kriging surrogate model was used) with the CFD evaluation for each iteration. An iteration within this method consists first of a training the surrogate model on the existing data, followed by a DE optimization using the surrogate model instead of the CFD evaluation, and a validation of the obtained best design by CFD. As can be seen, during the first iteration the surrogate model does not represent reality well, such that the DE optimization results in a design for which the surrogate model predicts a very large reduction in pressure drop. This is however not confirmed by the CFD validation, which shows that in fact a much larger pressure drop is obtained. This failure is added to the database after which a new iteration starts, consisting of retraining the surrogate model, optimizing the shape using the updated model and again verifying the result by CFD. As can be clearly seen, the surrogate model still overpredicts the reduction in pressure loss, but this time the prediction is already much closer to reality. Through adding the previous design to

the database, the system has learned valuable information preventing the optimization algorithm to search further in this wrong direction. The newly found design is added again to the database after which a new iteration starts. Gradually the difference between the surrogate model and CFD prediction is reduced until the accuracy of the surrogate predictions are confirmed by CFD. Similar to the previous study a 36% reduction in pressure drop could be achieved, although in the present case a 3D CFD computation was used.

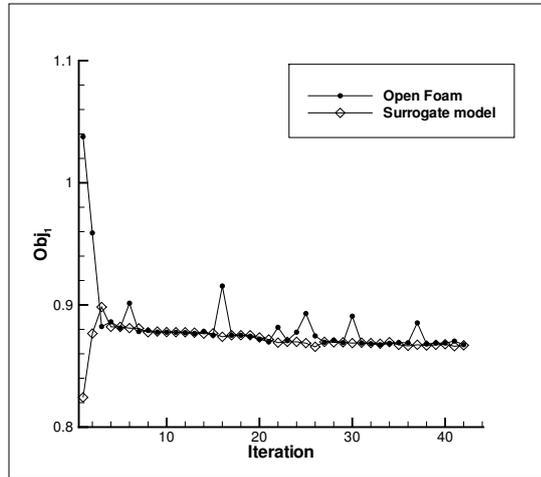


Figure 4: Convergence history of surrogate model assisted EA

Prior to the optimization a total of 65 designs were analyzed by CFD to have an initial training set for the surrogate model. With an additional 40 calculations needed to find the optimum, only about 100 CFD computations are needed, which is an order less than for the DE without surrogate model assistance.

In 5 the optimal shape is shown. Careful analysis demonstrated that the reduction in pressure drop was achieved through a suppression of the separation on the inward surface of the bend. This was achieved by reducing the curvature near the wall, hence decreasing the velocity gradient normal to the wall and reducing the adverse pressure gradient in the second half of the inward surface.

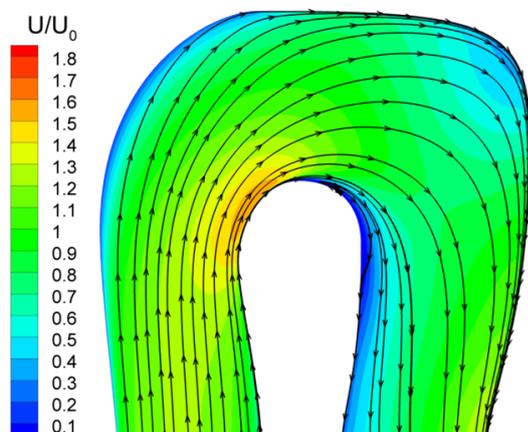


Figure 5: Optimal shape of the U-bend for minimal pressure loss

4.3 Multi-objective EA Assisted by Surrogate Model

So far only the pressure objective (1) has been minimized. The U-bend in the present work however serves to cool down a turbine blade, and as explained in the introduction an increased heat transfer is an additional aim. Especially the tip of the blade is a critical area which may benefit from a better cooling. Therefore the objective expressed by 2 is introduced. Both objectives are conflicting and need a multi-objective optimization to obtain the optimal solution.

In 6 the result of the optimization is summarized. It shows the total pressure drop versus the heat extracted for all 220 analyzed geometries. The baseline geometry consisting of the circular U-bend is indicated by a square, while the 65 samples generated for the initial database are represented by black dots. It is already apparent that these initial geometries perform better with respect to the total pressure drop objective, and most samples also perform better in the heat objective compared to the baseline.

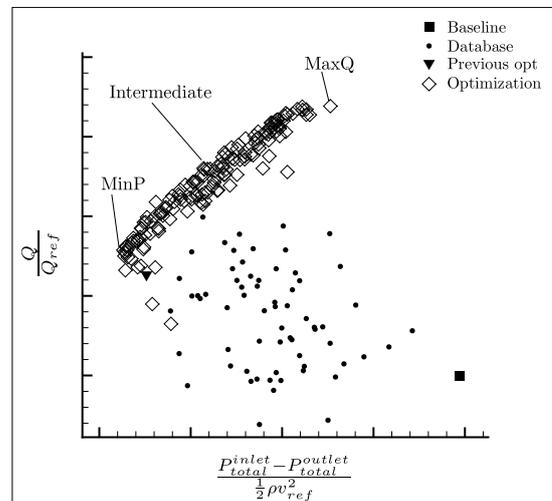


Figure 6: Results of the optimization plotted in the objective space

The samples generated during a total of 32 iterations of the optimization phase are represented by diamonds. All of them are generated in the region of interest, i.e. with high heat transfer and low pressure drop. A clear Pareto front is formed, for which one cannot improve one objective without worsening the other. This clearly indicates that pressure loss and heat transfer are conflicting requirements, i.e. a physical mechanism is responsible to increase one and at the same time decrease the other.

Three candidate solutions are identified as “Min” which has the lowest total pressure drop, “MaxQ” which has the highest heat transfer, and “Intermediate”, which is in between both extremes. The performance of all three Pareto optimal geometries is summarized in 1. Finally, the optimal solution found during the single objective optimization, as presented in the previous section, is plotted as a gradient symbol. Although this optimization was not targeting any heat transfer objective, it improved the heat transfer compared to the baseline, as was also found during experimental validation.

In 7 the shapes corresponding to the three identified candidates are shown. The geometry with lowest pressure drop (“MinP”) resembles very closely the shape of the single-objective optimum (see 5). The increase in heat transfer by going to “MaxQ” is obtained by increas-

Table 1: Objectives of the trade-off configurations

	Obj_1	Obj_2
Baseline	1.22	1.00
MinP	0.84	1.08
Intermediate	0.93	1.13
MaxQ	1.07	1.17

ing the curvature in the external wall in the first 90 degrees and by increasing the internal wall width. Both actions increase the pressure loss and transform the smooth configuration into one that resembles closer and closer a sharp u-bend configuration. Similar to what was found by Liou and Chen [10], a thicker divider wall is beneficial for the losses. Geometries with low pressure loss tend to have a smooth curvature change, and successfully suppress separation by increasing the radius of the turn and by carefully decelerating first and then accelerating the mean flow. As a consequence, less secondary flow motion is present and reduces the heat transfer potential. Geometries with high heat transfer on the other hand contain rapid changes of curvature and resemble close to sharp u-bends. Heat transfer is enhanced due to the impingement of the flow near the external wall, however increasing the losses.

The computational cost of the multi-objective optimization is with its 220 CFD evaluations slightly larger than the single-objective optimization of section 4.2. It however needs to be noted that for each of the 32 iterations 5 individuals need to be analyzed which is performed in parallel. This allows for a faster completion than the single-objective optimization, for which only 1 design is evaluated per iteration.

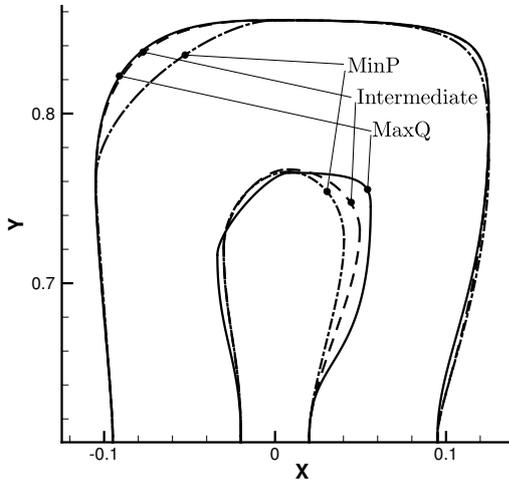


Figure 7: Comparison of the trade-off shapes

4.4 Gradient Based Optimization

The same 2D single-objective optimization as performed in section 4.1 has been repeated with a gradient based optimization method, although with a different parameterisation as explained in section 2.2. The gradient has been computed using the continuous adjoint approach implemented in OpenFoam. In 8 the optimal shape is shown compared to the initial shape.

A comparison of the optimal shape from EA based optimization algorithms to the best performing design obtained by the steepest-descent method reveals that both

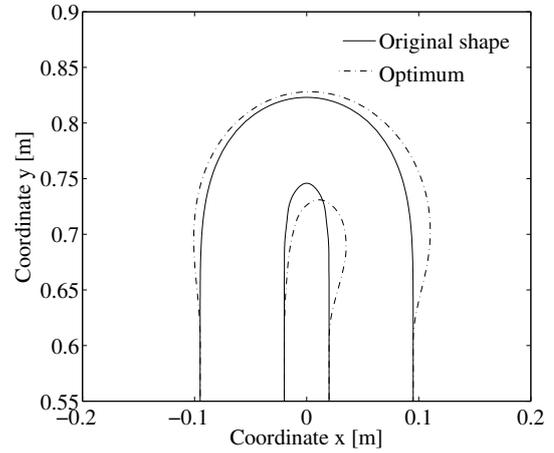


Figure 8: Comparison of baseline and adjoint optimization shape

U-bends exhibit similar geometrical features leading to a strong reduction of the total pressure drop. Both configurations feature an increased duct section in the first part of the bend resulting in a limited acceleration of the flow around the bend tip. In combination with increased radii of curvature along the internal and external walls, the reduced flow velocity leads to reduced centrifugal forces reducing the tendency of the flow to separate. In addition, the convex inner wall along the second leg of the bend is deformed such that it fills the space which is occupied by the separated flow in the original geometry. While present in the gradient-free optimized shape, this feature is even more pronounced by the gradient-based method. Considering the different geometry parameterisation resulting from the necessity to limit the number of design parameters for gradient-free optimization, the similarity of the optimal U-bend shapes obtained by differential evolution and steepest descent represents an unprecedented finding. This remarkable result demonstrates that the underlying objective function in the present case does not pose a multimodal problem as often assumed for engineering optimization problems. Consequently, both gradient-free and gradient-based optimization methods detect the global optimum demonstrating that the concern of getting trapped in a local minimum is of no relevance for the application of the latter. Therefore, by using a computationally efficient gradient-based optimization procedure a globally optimal U-bend shape is provided after only 30 design iterations where succeeding flow field computations benefit from previously converged solutions. The computational cost is thus almost an order of magnitude less than the EA, and this for a 4 times larger design space.

4.5 Experimental Validation

The baseline geometry consisting of 2 circular arcs and the shape shown in 5 has been investigated experimentally. In terms of global performance, 2 summarizes the experimental obtained improvement and compares them to the numerical predictions. The agreement between experiments and calculations is good, and the improvement in aerodynamic performance (both measured as well as predicted) is very significant.

Detailed PIV measurements have however also been performed and reveal a small recirculation bubble, not present in the numerical result. 9 shows the obtained velocity field, which can be compared to 5. It clearly

Table 2: Aerodynamic performance of the investigated U-bend configurations.

	ΔP baseline [-]	ΔP optimized [-]	gain [%]
Exp.	1.03 ± 0.03	0.65 ± 0.02	36.2 ± 3
CFD	1.01	0.63	37.6

demonstrates the limitations of the $k-\epsilon$ turbulence model in predicting flow separation in regions of adverse pressure gradients. Despite the differences in the flow details, however, the model allowed to predict well the global trends and combined with an optimization algorithm provides an extremely efficient methodology to improve the shape of the U-bend.

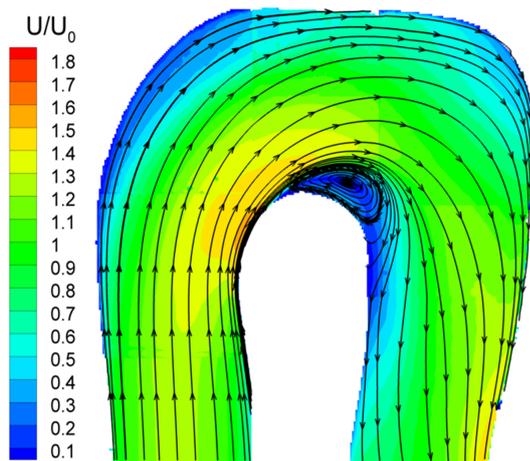


Figure 9: Mean velocity from PIV in the optimized geometry at mid height

5 Conclusions

An overview was given of different studies attempting to improve the performance of a U-bend for internal cooling channels. It was shown that all methods lead to shapes with similar features, in which the curvature of the inner wall has been reduced to limit the velocity gradient across the passage. When heat transfer is introduced next to the pressure losses as a second objective, several trade-off solutions can be found. The physical process behind the conflict between both objectives is due to the secondary flow motion. To increase heat transfer, a stronger secondary flow motion is desired, which can be introduced by a smaller curvature, however increasing the mixing losses and hence increasing the pressure losses.

Comparison between the different optimization methods demonstrates that the use of surrogate models can drastically reduce the required number of CFD evaluations from 4000 to 100 only. Comparing further the surrogate model assisted EA with the gradient based optimization, it was found that similar shapes were obtained despite the fact that the gradient based method departed from a separated initial design. It is often believed that engineering problems facing separation represent a multimodal character, for which gradient based optimization algorithms can get trapped in local optima.

In the present study however, results indicate that no such problems were present and seem to further feed the discussion as to which many engineering problems are unimodal of nature although easily thought multimodal

Finally, an experimental validation has proven the effectiveness of the optimization approach. In terms of global performance, the numerical predicted reduction in pressure losses was confirmed within measurement accuracy. Detailed PIV measurements however reveal a small separation which was not captured by CFD. It demonstrates that still further improvement should be possible, however beyond the capability of RANS approaches with their restriction on turbulence modeling.

Acknowledgment

The author would like to acknowledge the many contributors in obtaining the presented results. Many thanks go to Jérémy Bulle, Timothée Vanderwielen, Filippo Colletti, Tony Arts, Jing Li, and Sebastian Willeke.

References

- [1] J. A. C. Humphrey, J. H. Whitelaw, and G. Yee, “Turbulent Flow in a Square Duct with Strong Curvature,” *Journal of Fluid Mechanics*, vol. 103, pp. 443–463, 1981.
- [2] S. M. Chang, J. A. C. Humphrey, and A. Modavi, “Turbulent Flow in a Strongly Curved U-Bend and Downstream Tangent of Square Cross-Sections,” *Physico-Chemical Hydrodynamics*, vol. 4, pp. 243–269, 1983.
- [3] D. J. Monson and H. L. Seegmiller, “Experimental Investigation of Subsonic Flow in a Two Dimensional U-Duct,” in *NASA report TM-103931*, 1992.
- [4] S. C. Cheah, H. Iacovides, D. C. Jackson, H. H. Ji, and B. E. Launder, “LDA Investigation of the Flow Development Through Rotating U-ducts,” *Journal of Turbomachinery*, vol. 118, pp. 590–596, 1996.
- [5] Open Foam, “Open Foam user guide,” tech. rep., 2010. See also URL <http://www.openfoam.com/docs/user/>.
- [6] S. B. Pope, *Turbulent Flows*. Cambridge University Press, 2000.
- [7] J. H. Holland, *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [8] I. Rechenberg, *Evolutionsstrategie — Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Stuttgart: Fommann-Holzboog, 1973.
- [9] C. Othmer, “A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows,” *Int. J. Num. Meth. Fluids*, vol. 58, pp. 861–877, 2008.
- [10] T.-M. Liou, Y.-Y. Tzeng, and C.-C. Chen, “Flow in a 180S Sharp Turning Duct With Different Divider Thicknesses,” *Int. J. Num. Meth. Fluids*, vol. 121, pp. 569–576, 1999.

MISSION AND SHAPE OPTIMIZATION USING MODEFRONTIER: APPLICATION TO BOOMERANG THROW

R. Russo¹, A. Clarich¹, C. Poloni², E. Nobile²

¹*ESTECO Spa, Trieste (Italy), www.esteco.it*

²*Dipartimento di Ingegneria e Architettura, University of Trieste (Italy), www.units.it*

Abstract

This paper illustrates a proof of concept of a multi-level optimization, involving parameters related to both geometry and mission control. The goal is indeed to find an optimal boomerang in terms of energy requested to throw the object, optimizing at the same time the shape of the boomerang (high level) and the throw parameters (low level). The optimization problem is accomplished by the modeFRONTIER[®] platform; the throw parameters optimization takes into account also how the parameters uncertainty affects the boomerang performance (return capability).

1 Introduction

Product design is always a multidisciplinary task and, although optimization can be applied both at system and component level, the impact and success of the procedure is higher if the objectives are posed at the highest possible level. For this reason we have chosen in this paper to illustrate how the design optimization software modeFRONTIER[®] can be employed in a sports related optimization problem, where both shape and mission parameters have to be considered, including as well the effect of uncertain parameters. The application consists in the optimization of the shape of a boomerang and of its throwing parameters (mission optimization). A boomerang, shown in 1, is a flying object apparently simple but particularly challenging for the complex physics modeling, since it involves:

- six degrees of freedom body dynamics;
- aerodynamics of rotating blades;
- personal capabilities of the thrower.



Figure 1: Example of a typical boomerang

The boomerang trajectory is obtained by a dynamic model integrated to CFD analyses to compute aerodynamic coefficients. To steer the complete optimization process modeFRONTIER[®] is coupled to CATIAv5[®] for the boomerang shape modification, to MATLAB[®] for

dynamic simulation, and to STAR-CCM+[®] for aerodynamic analyses. Moreover, dedicated RSM (Response Surfaces Methods) available in modeFRONTIER[®] are used to extrapolate the aerodynamic coefficients as a function of the boomerang angle of incidence and velocity, as required by the dynamic model, allowing a reduced number of CFD analyses for each geometric configuration. Different design simulations are therefore executed automatically by modeFRONTIER[®] following a dedicated optimization strategy, until the optimal geometry of the boomerang is found accordingly to the specified requirements, such as minimum energy for the throw, and desired accuracy in returning [1]. In addition, a further optimization approach is proposed, which takes into account the uncertainties of the throw parameters. The methodology, which takes the name of Reliability-based Design Optimization, aims to find the optimal set of throw parameters under all the possible operating uncertainties, and is totally implemented in modeFRONTIER[®], taking the benefits of an innovative procedure based on the performances distribution estimation by Polynomial Chaos expansion.

2 Equations of the Boomerang Motion

Considering that a boomerang spins fast, it is possible to write the so called smoothed boomerang equations [2–6], 1, in which the different quantities (velocities, angles, forces) are time-averaged over a boomerang rotation:

$$\begin{aligned}
 \dot{\omega}_z &= \frac{T_z}{I_z} \\
 \dot{V} &= \frac{1}{m} (-F_x \cos \Psi - F_z \sin \Psi) \\
 \dot{\Psi} &= \frac{1}{mV} (F_x \sin \Psi - F_z \cos \Psi) + \frac{T_x}{I_z \omega_z} \\
 \dot{\vartheta} &= \frac{1}{I_z \omega_z} (-T_y \cos \psi - T_x \sin \psi) \\
 \dot{\phi} &= \frac{1}{I_z \omega_z} \frac{1}{\sin \vartheta} (-T_y \sin \psi + T_x \cos \psi) \\
 \dot{\psi} &= -\frac{F_y}{mV \cos \Psi} \left(-\tan \Psi \frac{T_y}{I_z \omega_z} - \cos \vartheta \dot{\phi} \right)
 \end{aligned} \tag{1}$$

where: I_z the maximum boomerang principal moment of inertia; V is the velocity magnitude of the boomerang center of mass; m is the boomerang mass; Ψ is the angle of incidence of the boomerang; ϑ, ϕ, ψ are the Euler angles of a xyz reference system partially fixed on the boomerang (such that the boomerang center of mass is always placed in the xyz origin, the z axis is always

directed as the maximum boomerang moment of inertia axis, and the projection of the boomerang center of mass velocity on the xy planes is directed as the x axis); ω_z the boomerang angular velocity around the z axis; $T_x, T_y, T_z, F_x, F_y, F_z$ are torque and force components, respectively, in the xyz reference system, basically due to the interaction between the boomerang and the air, and to the gravity force. The gravity force can be expressed in the xyz reference system as:

$$\vec{F}_g = -mg (\sin \vartheta \sin \psi, \sin \vartheta \cos \psi, \cos \vartheta)$$

The absolute velocity (and hence the position) of the boomerang center of mass can be found as function of the previous parameters by:

$$\begin{aligned} \dot{X} &= V (-\cos \Psi (\cos \psi \cos \phi - \sin \psi \sin \phi \cos \vartheta) \\ &\quad - \sin \Psi \sin \phi \sin \vartheta) \\ \dot{Y} &= V (-\cos \Psi (\cos \psi \sin \phi + \sin \psi \cos \phi \cos \vartheta) \\ &\quad + \sin \Psi \cos \phi \sin \vartheta) \\ \dot{Z} &= V (-\cos \Psi \sin \psi \sin \vartheta - \sin \Psi \cos \vartheta) \end{aligned} \quad (2)$$

The equations of motion can be integrated numerically using a high order Runge-Kutta method, once the initial conditions are provided and the forces and torques are available at any time step. A candidate boomerang trajectory can therefore be simulated through the following steps:

1. for a certain number of Ψ and U pairs (where $U = V/(\omega_z a)$, with a distance between the boomerang center of mass and the farthest boomerang point from the center of mass) the corresponding non-dimensional values of \vec{F} and \vec{T} are computed by CFD simulations: a dimensional analysis can prove indeed that \vec{F} and \vec{T} depend only on Ψ and U for a given boomerang geometry and for a Reynolds number range typical of boomerang flights;
2. response surfaces for $\vec{F}(\Psi, U)$ and $\vec{T}(\Psi, U)$ are built;
3. the equations of motion are integrated starting from the given initial conditions and using the response surfaces computed previously to express forces and torques at any position and time step.

The trajectory of the boomerang is affected of course by the initial conditions, namely by the way the boomerang is thrown. Four launching parameters are considered (they will be automatically tuned for each candidate boomerang by the optimization methodology described in section 5):

- **V**: initial boomerang translational velocity;
- **Spin**: initial boomerang spin;
- **Aim**: angle between the initial boomerang translational velocity and the horizontal plane;
- **Tilt**: angle between the initial boomerang rotational plane and the vertical axis (0° tilt corresponds to a vertical boomerang plane of rotation).

3 Boomerang Parameterization

The boomerang geometry chosen for the optimization will be the classical two arms “V” and “Ω” shape type [4]. The most important parameters that affect the boomerang behavior are linked to the blades profile, to the angle between the two arms and to the dihedral of the arms. A total number of 9 input parameters has been defined.

3.1 Blade Profiles

Changing the profile by playing with the angle of attack and cut on the top of the leading and trailing edge can change a lot the lift provided by the arm. The lift in particular affects the turn capability of the boomerang (precession effect).

The arc blades are in general designed with a positive angle of attack; this helps the boomerang plane to lay down and to float in air.

For the parametric boomerang geometry a flat bottom airfoil has been chosen. The blades profile are built by a Bezier parametric curve [7], with 4 control points. The profile shape is modified by changing vertical and horizontal position of the Bezier control points. In this way it is possible to change the angle of attach and the thickness of the blades (see Fig 2). In order to reduce the number of parameters, the profiles of the leading and of the trailing arm are controlled by the same parameters. In particular the vertical position of the trailing arm is set as a fraction of the vertical position of the leading arm.

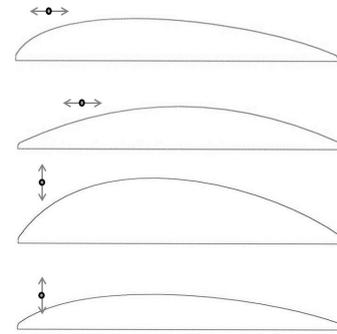


Figure 2: Effect on blade profile of Bezier control points

3.2 Dihedral Angle

Boomerangs arms usually have a positive dihedral angle that is about 10° - 15° ; the dihedral angle affects both the lift and the lay down velocity of the rotation plane, keeping practically unchanged the mass of the boomerang. The boomerang parametric model is provided with the two parameters α and d that allow to change the dihedral angle by removing a small amount of material at the boomerang arms tips, as illustrated in Fig. 3 (Fig.2). The α parameter is basically the stabilizer’s angle of attack.

3.3 Angle Between Arms

This angle usually ranges between 70° and 140° . In fact this parameter has a great effect on the boomerang stability. The length of the arms is fixed to keep a constant overall size of the boomerang.

4 Aerodynamic Forces Computation by CFD

The approach we considered consists in using two reference systems: one external and inertial, the other fixed with respect to the boomerang and having its origin placed in the boomerang center of mass. Also two domains and two grids are used: the first spherical having

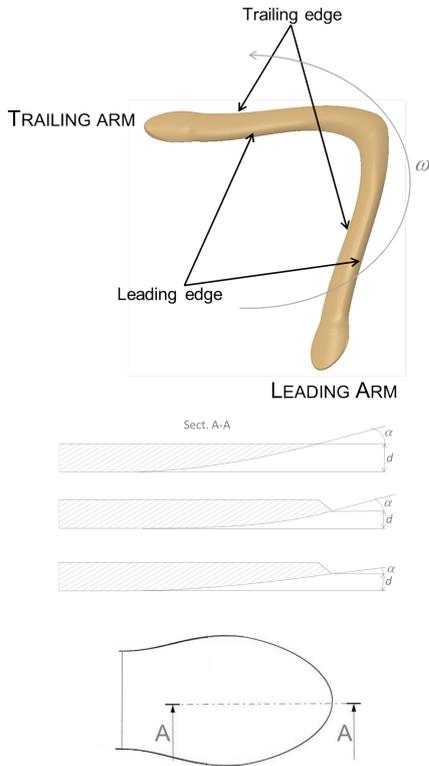


Figure 3: Leading and trailing edges; dihedral angle.

its origin placed in the boomerang center of mass and associated to the boomerang reference system, and the second corresponding to an external parallelepiped shape associated to the external reference system.

The internal spherical domain is provided with a rotational velocity around an axis normal to the boomerang plane and passing through the boomerang center of mass. The information between the two domains is exchanged by an interface boundary that allows to interpolate the field values.

In Star-CCM+[®] a polyhedral mesh is defined within the sphere around the boomerang, with prisms layers at the boomerang walls, and an hexahedral mesh is defined in the rest of the domain (Fig.3). A mesh size of about 2.5×10^6 cells has been defined, this being a good trade-off between accuracy and computational efforts.

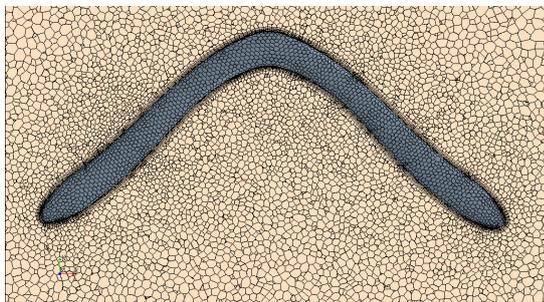


Figure 4: Particular of a mesh section

The two-equations RANS SST (Shear Stress Transport) turbulence model, with wall functions, is chosen and a segregated pressure based solver with constant density is employed. 5 depicts the pressure field on a boomerang surface at consecutive instants during a rotation. It is possible to see how the pressure force on each

arm changes significantly during the rotation according to the relative position of the blades with respect to the translational velocity.

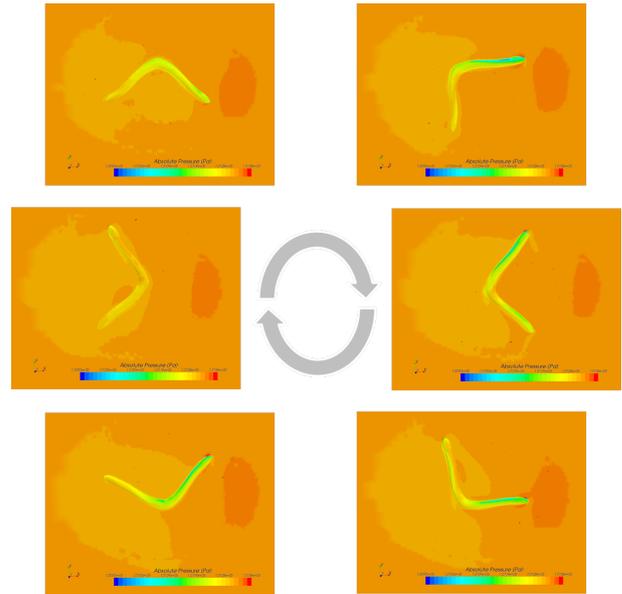


Figure 5: CFD results on different revolution frames

At the end of the numerical simulation (for a given Ψ, U pair) the averaged forces and torques acting during the rotation are computed and then the corresponding \vec{F} and \vec{T} are available.

5 Process Flow Automation in modeFRONTIER[®]

The whole process needed to evaluate and optimize the performances of the boomerang has been completely automatized through the software modeFRONTIER[®] [8]. In this modular environment, the complete process flow is defined by the user, that can select among several available optimization algorithms, including Genetic and Evolutionary Algorithms, Game Strategies, Gradient-based Methodologies, Meta-Models and Robust Design Optimization. The optimized boomerang is found automatically through the following steps:

1. modify the boomerang CAD model parameters;
2. obtain the updated geometry (.stl file) from the CAD and transfer it to the mesh generator;
3. launch the mesh generator to build the computational grid;
4. launch different CFD simulations using the same mesh prepared as above varying U and Ψ parameters for an appropriate number of samples; for each U and Ψ pair the corresponding forces and torques \vec{F} and \vec{T} are obtained ;
5. use the set of simulations computed in step 4. as training set for response surfaces to obtain $\vec{F}(\Psi, U)$ and $\vec{T}(\Psi, U)$ over the whole range of variation of Ψ, U ;
6. pass the response surfaces and the boomerang inertia data (from the CAD) to a script designated to

integrate numerically, using the MATLAB® *ode45* solver, the differential equations described in section 2, obtaining finally the boomerang trajectory;

7. run an internal optimization for the given configuration to tune the four launching parameters (by minimizing the distance between throwing and arrival position);
8. the main multi-objective algorithm assesses how good the trajectory is with respect to specified objectives (total energy needed for the throwing to be minimized);
9. the steps (1.-8.) are repeated automatically by the algorithm until one, or more, optimal configurations are obtained.

The modeFRONTIER® workflow is shown in 6. In particular, on the top we find the nodes (green subsystem) that define the range of variations of all the geometrical parameters, then the process flow (black line) starts with the interfaces to select the optimization algorithms and set their options, to continue with the CAD direct interface that allows to automatically update the geometric model at the variation of the parameters, obtaining as results the updated *.stl* model, which is transferred to the following script node used to run the mesh generator to create the mesh for the proposed geometry.

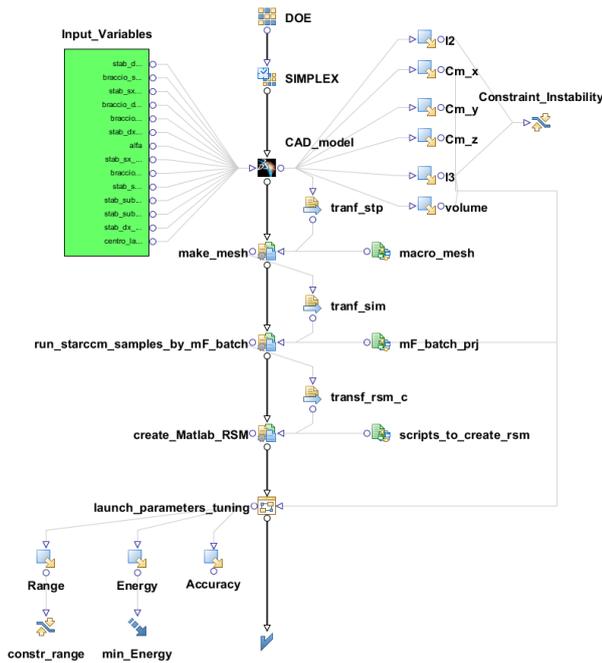


Figure 6: modeFRONTIER main workflow

The mesh (*.sim* file) is then transferred to the following application node, which basically launches in batch mode another modeFRONTIER® project file, which runs a set of CFD computations on the same mesh varying U and Ψ parameters, as described at point 4. above. The output from the internal modeFRONTIER® project is a Response Surface (RSM) or Meta-model, based on the available training set, which is able to extrapolate $\vec{F}(\Psi, U)$ and $\vec{T}(\Psi, U)$ over the whole range of variation of the two parameters, as illustrated, for the x and z components of \vec{F} , in 7. The algorithm used for the RSM

training is *Kriging* [9], and the model is automatically exported as a C script, which can be read by MATLAB®.

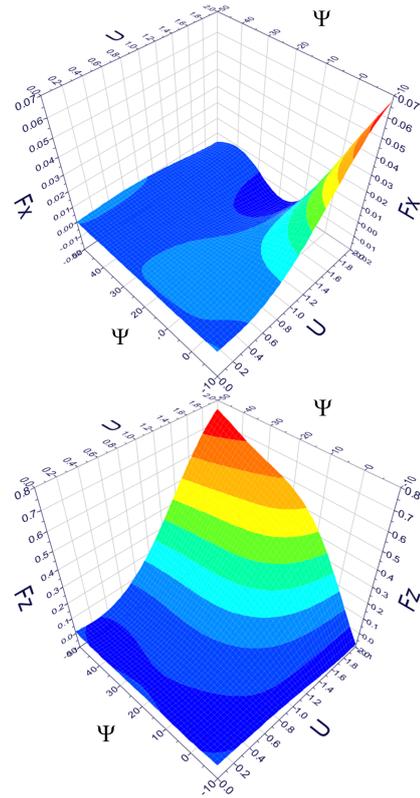


Figure 7: Response surfaces of x and z components of the boomerang aerodynamic force $\vec{F}(\Psi, U)$

The last application node in the process flow is therefore another modeFRONTIER® project node, called *launch_parameters_tuning*. This node actually runs another optimization project in batch mode, for which the input variables are the four throw parameters described in section 2, the boomerang shape is fixed and the objective is defined by the minimization of the distance from the arrival position and the launching position. For this purpose, a fast mono-objective algorithm is used (Simplex), and the project just executes, for each set of launching parameters, a MATLAB® script through the corresponding direct interface, that basically drives a Runge-Kutta integration to compute the boomerang trajectory (retrieving the needed $\vec{F}(\Psi, U)$ and $\vec{T}(\Psi, U)$ values for each integration time step directly from the Response surface available for each boomerang geometry). The final outcome of the modeFRONTIER® Batch node in the main process flow for each boomerang geometry is therefore its tuned trajectory, whose performances are to be optimized in the external loop. For this purpose, from this node the following outputs are extracted:

- **Range:** this is the maximum distance reached by the boomerang during its flight; it has just been considered as a constraint in the optimization, to penalize configurations having too small range;
- **Accuracy:** this is the difference between the position from which the boomerang is launched and the position where the boomerang returns (optimized by the internal loop as described above for each boomerang candidate solution);

- **Energy:** this is the energy - translational plus rotational - necessary to throw the boomerang, that is a quantity to be minimized (to reduce the effort for the thrower).

6 Optimization Strategy and Optimization Results

Several tests were performed to find the proper number of simulations requested to create enough accurate response surfaces. It has been found that a matrix of 12 points guarantees an error of approximation less of 5%, and this was the size of the training set finally selected. This means that each boomerang trajectory computation needs 12 CFD simulations. For this reason a classical multi-objective optimization algorithm that may require hundreds of designs evaluations is not practically feasible, therefore a different strategy, based on *Game Theory (Hierarchical Games)*, has been chosen [10].

As indicated in the previous chapter, two different objectives - returning accuracy and throw global energy - have to be considered, but actually any candidate solution is first optimized by the internal workflow in order to tune the launching parameters (*follower player*), then the found optimal solution is evaluated by the external optimization workflow that handles the energy objective minimization by changing in the proper way the geometrical parameters (*leader player*) [11]. Note that for both the internal and external optimizer the same modeFRONTIER® algorithm, Simplex, has been used due to its efficiency to solve single-objective problems in few iterations.

Fig. 8 reports the global results of the optimization process, in the space of the objectives and constraints considered.

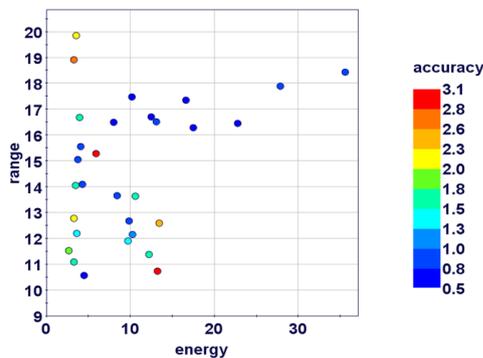


Figure 8: Optimization results

In particular, for each design, the abscissa reports the throwing energy (Joule), the ordinate indicates the range (meters), and the color of the bubbles reports the returning accuracy for each design (distance in meters).

At the end of the process, one of the optimal boomerang configurations has been chosen, and its geometry and trajectory are also depicted in Fig. 9. The energy required to throw the boomerang is 3.5 J, the ratio of rotational to total energy is only 7% which corresponds to an initial spin of about 4 Hz and an initial translational velocity equal to 15 m/s. The tilt angle is 0°, while the aim is about 20°. This set should make the boomerang throwing pretty easy, with a range of 14.5 m.

The optimized boomerang, which was realized using epoxy resin and carbon fibers on a 3D printed mould, is depicted in 10. It proved to be extremely effective and

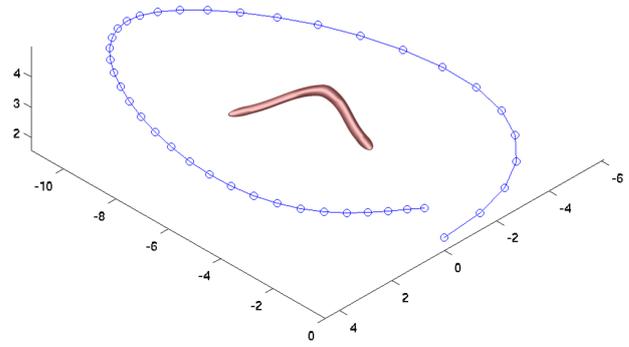


Figure 9: Optimal boomerang configuration and trajectory

accurate, and yet simple to throw even by the authors of this study, as shown in fig. 11, which have no particular experience or expertise in this matter.



Figure 10: The boomerang prototype obtained using epoxy resin and carbon fibers on a 3D printed mould



Figure 11: One of the authors enjoys throwing the optimized boomerang

7 Reliability-based Design Optimization

Reliability-based Design Optimization (RBDO) is achieving more and more consensus in the industrial design community. In fact, most of the industrial processes are permeated by uncertainties: the manufactured product is generally different, from a geometric point of view, from the product design because of the dimensional tolerances and, more frequently, because the working point is not fixed, but is characterized by some fluctuations in the operating conditions.

Input variables		Range	Uncertainty (std. dev.)
Velocity (V)	[m/s]	[5-30]	2
Spin (ω_z)	[Hz]	[0-10]	1
Aim angle (ψ_0)	[°]	[0-30]	2
Tilt angle ($90^\circ - \vartheta_0$)	[°]	[0-50]	2
Objectives		Goal	
Returning distance RD	[m]	Minimize 99-ile RD	
Range	[m]	Maximize avg. value	

Table 1: Input variables and objectives considered for the reliability-based optimization. The subscript “0” indicates the initial condition

This uncertainty is commonly transferred to the performance of the system, which cannot be determined with an exact and single value, but which is better described by a statistical distribution of results. In this environment, a frequent industrial requirement is the satisfaction of constraints or limits, which should be achieved for a specified percentage of the performance distribution, or for which the percentage of solutions not satisfying the limits (failure probability) must be minimized as much as possible, to improve the reliability and quality of the product.

To deal efficiently with a reliability-based design optimization problem of industrial relevance, we propose an innovative methodology, fully implemented in the modeFRONTIER[®] software, which conjugates accuracy and reduced number of needed evaluations. The methodology is based on the application of *Polynomial Chaos Expansion* [12], which means that for each candidate design proposed by the optimization algorithm, a small sampling set is evaluated by varying from the nominal values the uncertain parameters accordingly to the given distribution, and then *Polynomial Chaos Expansion* is used to evaluate the output performances distribution [13]. More specifically, the polynomial coefficients are, at this point, used to evaluate the complete cumulative distribution function of the performances of the design [14], from which it is possible to retrieve accurately the failure probability for the prescribed limits/constraints of the problem, on which the optimization objectives are defined.

As an application case, we extend the boomerang design optimization problem to a reliability-based optimization concept, since we want to optimize the boomerang throw, subjected to many uncertainties like throw angles, velocity and spin, with the purpose of minimizing the percentage of throws which do not return accurately, i.e. the failure probability. For simplicity reasons, we assume that the geometry of the boomerang is given, as the optimal one found by the deterministic optimization described in chapter 5, and corresponding to the one depicted in 9. The input variables for the reliability optimization problem are therefore just the ones related to the boomerang throw (already introduced in sec. 2), as expressed in table 1 below with their range of variation and their uncertainty parameters (considering a Normal distribution for each one).

The range of variation includes all feasible values that can be obtained by a common throwing, and the uncertainties (to be quantified just by their standard deviation assuming a Gaussian distribution) take into account any possible random variation that a generic thrower (not

Input variables		Optimal range	Optimal return
Velocity (V)	[m/s]	21.6	21.7
Spin (ω_z)	[Hz]	4.98	4.92
Aim angle (ψ_0)	[°]	4.2	4.2
Tilt angle ($90^\circ - \vartheta_0$)	[°]	20.1	7.2
Objectives			
Returning distance RD	[m]	8.5	2.9
Range	[m]	33.4	21.7

Table 2: Reliability-based optimization results

necessarily a professional one) might produce. The optimal set of launching parameters will therefore give the lowest failure probability taking into account the random perturbation due to the thrower behavior or other random events. As optimization objective, we arbitrarily consider the minimization of the 99-th percentile of the return distance: this means that 99% of throws will return to a distance lower or equal to this value, and the purpose is to minimize it. For a more efficient result, we add a second objective, which is the maximization of the range of the throw; in this case, since there is not any particular constraint to achieve, we just consider the average value of range. To solve the RBDO problem here described, modeFRONTIER[®] has been applied again, slightly modifying the workflow illustrated in 6. In this case in fact only the input variables related to launching parameters and application nodes related to trajectory simulation (MATLAB[®]) have been kept, since the boomerang geometry and the corresponding aerodynamic coefficients are constant and equal to the one computed for the optimal design found in section 6. A proper multi-objective optimization algorithm available in modeFRONTIER[®] is at this point selected (NSGA-II [15]), and the optimization is executed automatically, running 10 generations of 16 designs each, for an overall number of proposed designs equal to 160. For each of these designs the *Polynomial Chaos Expansion* is applied to evaluate the distributions performance, and for this purpose, considering 4 uncertainties and a Polynomial degree equal to 2, 15 sampling points are needed for each design.

As a conclusion, after an overall number of 2400 different evaluations (which is rather a low number for a reliability-based multi-objective optimization problem with 4 uncertainties), the results reported in table 2 are obtained.

It is worthwhile to emphasize that the CPU time requested for each design evaluation is of the order of one second, so that the entire RBDO problem took less than one hour.

Two optimal results have been selected and reported in the table: the first one gives optimal performances for the range, while the second guarantees the most accurate throw: in 99% of the cases the returning distance will be less than 2.9 m. Finally, fig. 12 reports the complete results of the optimization: in abscissa we have the 99-ile of the return distance and in ordinate the average range, while the color represent the nominal value of the return (or arrival) distance.

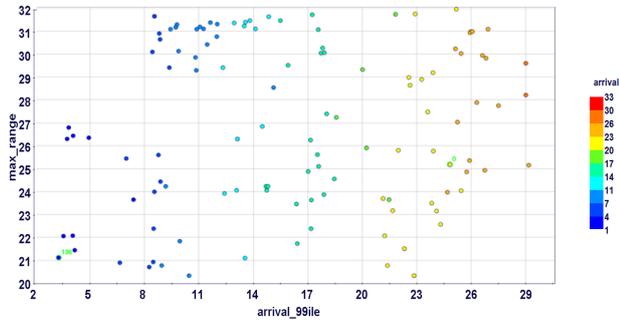


Figure 12: Reliability-based multi-objective optimization results: max range vs arrival 99-ile

8 Conclusion

In this paper we have described an automatic and efficient methodology for the multi-objective optimization of a boomerang throwing, as an interesting benchmark and proof of concept to illustrate the multi-objective and multi-disciplinary capabilities of the optimization environment modeFRONTIER[®], in particular for what concerns mission optimization of practical relevance. In addition, the concept of reliability-based multi-objective design optimization has been introduced to the same application, allowing to find the optimal set of launching parameters for the optimized boomerang shape, which guarantees at the variation of uncertain parameters the 99% of throws returning back at a minimum distance with the maximum average range distance.

References

- [1] R. Russo, A. Clarich, E. Nobile, and C. Poloni. Optimization of a Boomerang Shape Using modeFRONTIER, AIAA Paper 2012-5489,. In *AIAA Aviation Technology, Integration, and Operations Conference*, 17-19 September 2012.
- [2] F. Hess. *Boomerangs, aerodynamic and motion*. Phd thesis, University of Groningen, 1975.
- [3] M.J. Hanson. *The boomerang book*. Penguin Books Ltd, England, 1974.
- [4] F. Hess. The Aerodynamics of Boomerangs. *Scientific American*, 219:124–136, 1968.
- [5] J.B. Mauro. *An introduction to Boomerangs*. Smith & Flannery Inc., Richmond, VA, 1974.
- [6] http://www.researchsupporttechnologies.com/boomerang_site/boomerang1.htm.
- [7] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Inc., San Diego, 1990. pp. 51-55.
- [8] www.esteco.com.
- [9] E. Rigoni and A. Turco. Metamodels for Fast Multi-objective Optimization: Trading Off Global Exploration and Local Exploitation. In *Eighth Int. Conf. on Simulated Evolution And Learning (SEAL-2010)*, pages 523–532, 01-04 December 2010.
- [10] J.F. Wang and J. Periaux. Search Space Decomposition of Nash/Stackelberg Games using Genetic Algorithms for Multi-Point Design Optimization in Aerodynamics. In *Domain Decomposition Methods in Science and Engineering (CIMNE 2002)*, pages 139–149, 2002.
- [11] A. Clarich and C. Poloni. Multi-objective optimisation in modeFRONTIER[®] for aeronautic applications, including CAD parameterisation, Evolutionary Algorithms, Game Theory, Metamodels and Robust Design. In *Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems – EUROGEN 2007*, pages 465–470, 11 - 13 June 2007.
- [12] G. Loeven, J. Witteveen, and H. Bijl. Probabilistic collocation: an efficient non-intrusive approach for arbitrarily distributed parametric uncertainties, AIAA Paper 2007-317. In *45th AIAA aerospace science meeting and exhibit*, 8 - 11 January 2007.
- [13] A. Clarich and V. Pediroda. Robust design applications with modeFRONTIER[®] applying NODESIM-CFD tools. NODESIM-CFD Workshop on Quantification of CFD Uncertainties, 29-30 October 2009. Brussels-Belgium.
- [14] A. Clarich, M. Marchi, E. Rigoni, and R. Russo. Reliability-based design optimization applying polynomial chaos expansion: Theory and applications. In *ECCOMAS Thematic Conference Multibody Dynamics 2013*, pages 947 – 956, 01-04 July 2013. Zagreb, Croatia.
- [15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.*, 6:182 – 197, 2002.

ADJOINT-BASED OPTIMISATION METHODS FOR AUTOMOTIVE APPLICATIONS

C. Othmer

Volkswagen AG, Group Research, Wolfsburg, Germany

Abstract

The adjoint method is commonly regarded as the tool of choice for gradient-based optimisation in computational fluid dynamics (CFD). While being well-established in the aeronautical sector, it has only recently started to enter the development processes of the automotive industry. From the perspective of Volkswagen Group Research, this article provides a concise status report on the achievements and remaining challenges of the development and application of adjoint-based optimisation methods in various automotive CFD disciplines.

1 Introduction

Pioneered by Jameson [1] for aeronautic applications in the 1980s, the adjoint method is now making its way into the automotive industry. At Volkswagen Group Research, adjoint development work started in 2006 with the implementation of a basic continuous adjoint for topological [2] and shape sensitivities [3] in OpenFOAM® [4]. Through the contributions of various academic and commercial partners – above all the teams around K. C. Giannakoglou (NTUA, Greece) and E. de Villiers (Engys, UK) – this basic adjoint implementation developed over the last almost 10 years towards a versatile continuous adjoint solver framework of industrial maturity (see [5, 6, 7] and references therein).

As sketched in 1, a number of different automotive optimisation disciplines are being tackled with this solver: topology and shape optimisation of ducted flows (e.g. airducts for cabin ventilation or engine ports), flow control and shape optimisation in external aerodynamics, and two areas where our adjoint research has just started: cooling and aeroacoustics. The article will present snapshots of each of these disciplines in order to give an impression of where the adjoint method stands today at

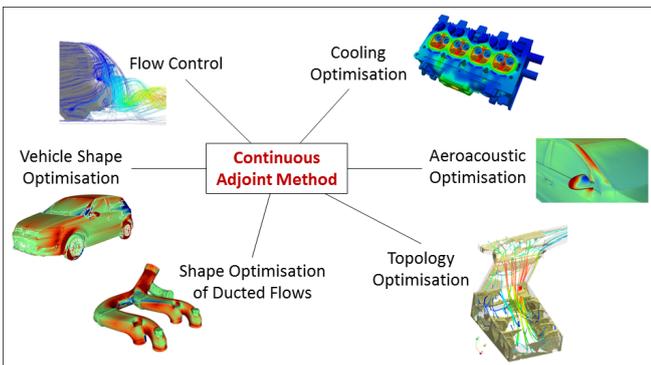


Figure 1: Automotive optimisation disciplines that are tackled with the adjoint method at Volkswagen Group Research

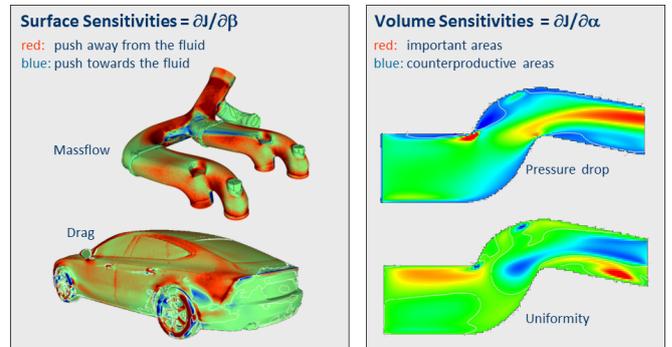


Figure 2: Examples of surface and volume sensitivity maps – the basis for shape and topology optimisation, respectively. The topological sensitivities are displayed for a cut through an S-shaped airduct

Volkswagen Group Research. Starting with topology optimisation for ducted flows, we will proceed clockwise in 1 in the order of decreasing method maturity.

2 Automotive Applications

At the heart of gradient-based optimisation is the computation of sensitivity maps, examples of which are depicted in 2: on the one hand, surface sensitivity maps, i.e. the gradient of the cost function J w.r.t. normal node displacements β , and on the other hand topological (or “volume”) sensitivity maps. The latter represent the cost function gradient w.r.t. changes of the Darcy porosity α of each individual volume cell, and indicate how essential the cell is for the fluid dynamic performance, thus forming the basis for CFD topology optimisation.

2.1 Topology and Shape Optimisation of Ducted Flows

Topology optimisation is a well-established concept in structure mechanics to create optimal lightweight structures that have to fit into a given installation space. It almost comes as a surprise that it took until 2003 before this elegant idea was transferred to fluid dynamics [8, 9]. Here, similarly to its structure-mechanical counterpart, the starting point is the entire installation space – the whole fluid domain is “flooded”. Based on the topological sensitivity $\partial J/\partial\alpha$, counterproductive areas of the flow domain are then identified and iteratively punished by adding some amount of Darcy porosity, ideally in a one-shot fashion. The optimal duct geometry is obtained at the end of the optimisation by collecting all cells with zero porosity [2, 7].

This method is very efficient for drafting duct designs from scratch. At Volkswagen, its most productive use to

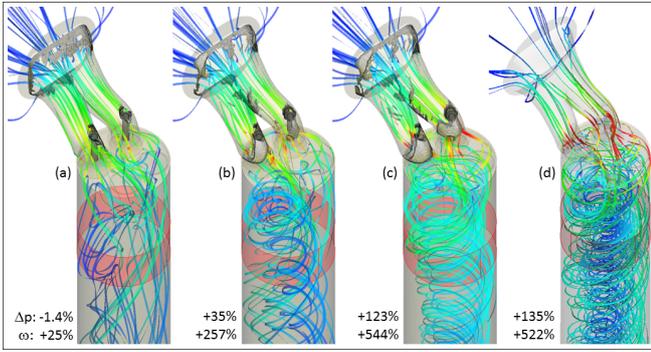


Figure 3: Intake port topology optimisation: The three figures (a)-(c) show the Pareto-optimal states for low, mid- and high swirl weighting, respectively. The grey areas within the intake port arms are the cells that were blocked during the topology optimisation. Note the obvious improvement of swirl ω from (a) to (c) at the cost of increasing pressure drop Δp . Figure (d) shows the flow field for a clean CAD model regenerated based on the results of case (c). For details we refer to [10]

date is made by the colleagues of the engine port development. On the one hand, they start from the available packaging space and “cut” the optimal exhaust port out of it. That is what topology optimisation is designed for and requires two loops of running the adjoint-based optimisation and subsequent manual CAD-reconstruction. On the other hand, they start from given port designs and make a multi-objective optimisation to explore the Pareto front of pressure drop vs. maximising swirl for intake ports (see the example in 3, [10]).

As topology optimisation is quite robust and straightforward to run, it was adopted quickly by the colleagues in the Technical Development and is used productively in a number of settings [7]. The main drawback of the present porosity-based realisation is that, by definition, it works on a stepped geometry, and until the very end there is no wall where a no-slip condition or a wall function could be applied, resulting in an inaccurate CFD solution. For a real geometry fine-tuning after topology optimisation, classical shape optimisation is therefore employed (4), by coupling the adjoint solver to a mesh motion tool developed by the team around K. U. Bletzinger (TU Munich, Germany, [11, 12]).

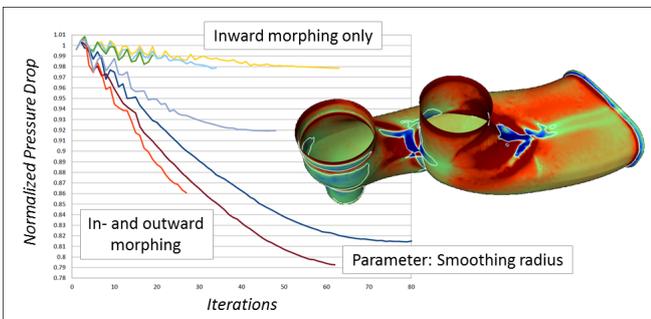


Figure 4: Exhaust port shape optimisation: Based on a pressure drop surface sensitivity map (colours like in 2 left), a one-shot shape optimisation was conducted with the morphing routines from TU Munich [11, 12], employing different smoothing radii and morphing directions (mesh kindly provided by F. Kunze)

2.2 Shape Optimisation and Flow Control in External Aerodynamics

The breakthrough for the adjoint method in external aerodynamics at Volkswagen was the application to the low-emission car XL1. Based on a RANS simulation and a subsequent adjoint run including the effect of adjoint turbulence [13], drag sensitivity maps were obtained (5). Among other things, the sensitivities clearly indicated that a rear spoiler would further improve drag. Accordingly, a one-shot optimisation was performed to obtain its optimal shape [5]. It converged after about 20 shape updates and resulted in a 2% drag reduction (6). Since the XL1 was aerodynamically nearly perfect already and the method only modified a small portion of the car surface, this improvement was rated as a success and provided credibility to adjoint-based one-shot shape optimisation as a useful tool in external car aerodynamics.

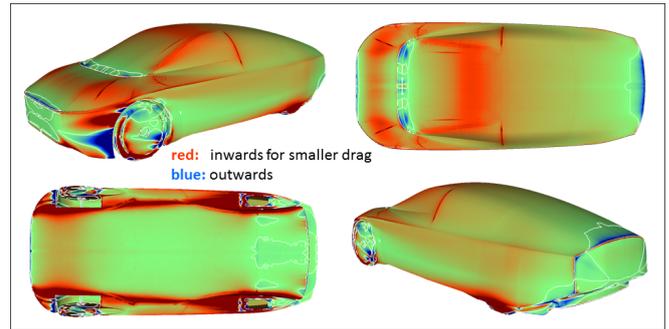


Figure 5: Drag sensitivity maps for the Volkswagen XL1: In red (blue) areas, an inward (outward) movement would result in drag reduction. White lines are the isocontours of zero sensitivity, and modifications in greenish areas have little effect on drag

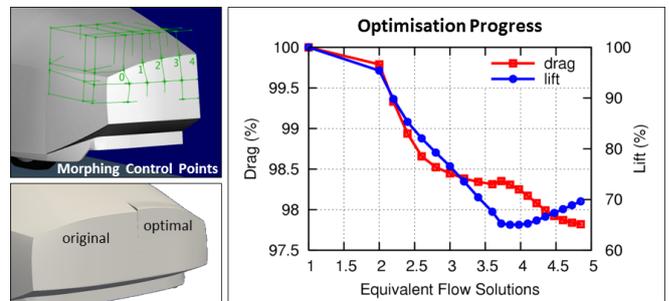


Figure 6: One-shot optimisation of the XL1 rear spoiler: Parameterisation (top left), optimisation history (right) and shape comparison (bottom left, details see [5])

However, the productive use of CFD for car aerodynamics has meanwhile moved from RANS to Detached-Eddy Simulation (DES, [14]). As DES is unsteady and computationally very demanding, it remains a challenge still today to also upgrade the adjoint towards being fully unsteady and capable of dealing with geometries of industrial complexity in acceptable turn-around times. Current research activities with academic and commercial partners address this issue [15]. As a temporal workaround, we are applying the following procedure to compute approximate sensitivities based on primal DES results: From the primal time-averaged DES simulation, we take the velocity field and solve a RANS turbulence model to compute a turbulent viscosity ν_t . The existing RANS adjoint solver is then executed based on the time-averaged primal DES velocity and the obtained RANS-

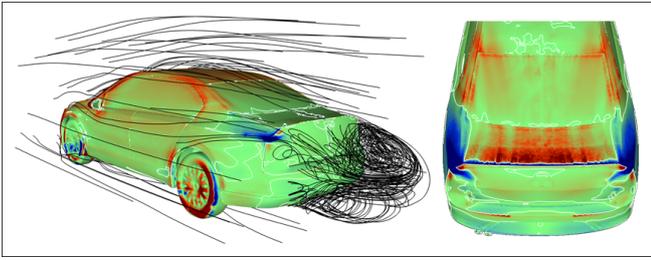


Figure 7: Example of a drag (left) and lift (right) sensitivity map for an intermediate design state of a sedan car. Note how the blue areas indicate suboptimal flow separation and thus potential for drag and lift reduction

ν_t . Quantitative accuracy of the sensitivities cannot be expected with such an approach – as compared to finite differences we are indeed far off. But via windtunnel measurements and numerical validations these approximate DES-based sensitivities have been shown to indeed provide an added-value as compared to pure RANS-based sensitivities [7]. This procedure has therefore become the standard way of computing DES-based drag and lift sensitivity maps for external aerodynamics at Volkswagen Group Research (see 7 for an example) – until our efforts in industrialising a fully unsteady DES adjoint solver are finished.

In external aerodynamics computations, the use of high-Reynolds turbulence models with wall functions is common industrial practice. The implementation of adjoint wall functions for the Spalart-Allmaras turbulence model by Papoutsis-Kiachagias et al. [16] was therefore another major step towards increased accuracy of aerodynamic sensitivity maps. 8 shows a comparison between drag sensitivities for the Volkswagen Polo obtained with the implemented adjoint wall functions and those based on the “frozen turbulence” assumption. While both approaches compute the same sensitivity sign for almost the entire surface of the car – i. e. deliver the same *qualitative* information – the latter tends to overestimate the absolute value of the sensitivity by more than one order of magnitude. For *quantitative* assessments, i. e. how much drag reduction can be achieved for a given deformation of the car shape, which is a crucial question within the car design process, the sensitivity map based on the frozen turbulence assumption proves to be unreliable and the usage of adjoint wall function is a must. It allows to cast sensitivity maps in handy units like “drag counts per *mm* displacement per *m*² surface area”, which are well-received in the aerodynamic development process by both the aerodynamic engineers and the stylists.

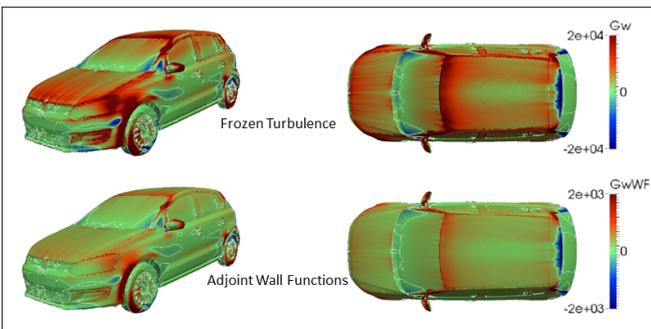


Figure 8: Influence of adjoint wall functions on drag sensitivities (note the different scales, details see [16])

Other than shape optimisation, another way of improving aerodynamic performance is obviously flow control. In order to design an efficient flow control system, it is important to know (i) where to place the jets for maximum efficiency, (ii) what kind of flow control – blowing or suction – is needed and (iii) how big the effect on drag will be. To answer these questions, at least for *steady* blowing or suction, the adjoint method is again the tool of choice: By computing flow control sensitivity maps, i. e. the sensitivity of the drag w.r.t. changes in the normal flow velocity on the car surface, we guide the positioning of blowing or suction jets and, via the magnitude of the sensitivity value, get an estimate of the achievable effect [17]. As an example, 9 shows such a sensitivity map for the Volkswagen XL1. A 1:4 model of this car was equipped with a blowing jet in the marked region and tested experimentally [18]. While the windtunnel measurements confirmed a drag improvement induced by the operation of the jets, the effect was, however, too small to run the jet system economically on this car. Adjoint-aided flow control research therefore continues.

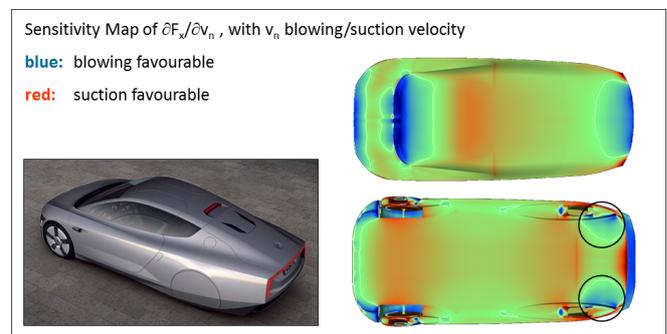


Figure 9: Flow control sensitivities for the Volkswagen XL1. In the experiments, blowing jets were applied in the areas marked by the black circles [18]

2.3 Optimisation of Cooling and Aeroacoustics

While the four optimisation disciplines discussed so far represent areas of productive use of the adjoint method in the Volkswagen Group, the last two applications of 1, adjoint-based optimisation w.r.t. cooling and aeroacoustics, are still under development.

The main motivation of extending the adjoint capabilities to include heat conduction is the increasingly challenging task of cylinder head cooling. Along the the rising trend of downsizing and turbocharging internal combustion engines for improved fuel efficiency, the dissipation of the emerging heat is actually becoming a bottle neck.

As a first step towards the inclusion of cooling objectives into the adjoint solver, the Reynolds analogy was exploited by defining the wall shear stress at the cooling interface as the objective function. Since this is a cost function defined solely in the fluid domain, its implementation in the existing adjoint solver was straightforward. Initial applications to cooling pipes showed promising results for this kind of flows [19]. The treatment of complex tasks like the removal of hot spots in the solid part of the cylinder head requires, however, the inclusion of conjugate heat transfer into the optimisation process. In cooperation with our partners from NTUA we are therefore currently extending the adjoint solver to also incorporate heat conduction in the solid and fluid domains as well as the heat transfer between them, with the aim

of optimising the shape of the fluid domain w.r.t. the temperature distribution in the solid domain.

On a similar level of industrialisation are our adjoint method developments for aeroacoustic optimisation. What has been successfully accomplished so far is the computation of approximate shape sensitivities for mirror noise sources [20]. Previous numerical studies had indicated that the turbulent kinetic energy close to the side window can serve as an indirect measure of the noise perceived at the driver's ear. Motivated by that, the adjoint solver including the adjoint turbulence model is used to compute shape sensitivities w.r.t. changes in the turbulent kinetic energy integrated over a volume adjacent to the driver's side window. While these sensitivities serve as a rough guidance for shape modifications that mitigate dominant noise sources, they are not capable of capturing the actual physics of mirror noise. Aeroacoustics is unsteady by nature, and its adequate treatment requires an industrialised unsteady adjoint in the first place. Work in that direction is in progress within the Marie-Curie Training Network AboutFlow [15].

3 Summary

As a result of a long-term collaboration with both academic and commercial partners, the continuous adjoint method has "hit the road" in the Volkswagen Group. For topology and shape optimisation of ducted flows as well as for flow control and vehicle shape optimisation in external aerodynamics it is in productive use already, and the extension of its capabilities to include the physics of conjugate heat transfer and aeroacoustics is underway.

Acknowledgement

The author is truly grateful to K. C. Giannakoglou and his team at NTUA, E. de Villiers and his colleagues at Engys, K. U. Bletzinger and his group from TU Munich as well as to J. D. Müller (Queen Mary University London) and the teams of FlowHead and AboutFlow. The support from many CFD colleagues of the Volkswagen Group is also gratefully acknowledged.

References

- [1] A. Jameson, "Aerodynamic design via control theory," *J Scientific Computing*, vol. 3, pp. 233–260, 1988.
- [2] C. Othmer, E. de Villiers, and H. G. Weller, "Implementation of a continuous adjoint for topology optimization of ducted flows," *AIAA-2007-3947*, 2007.
- [3] C. Othmer, "A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows," *Int J Num Meth Fluids*, vol. 58, pp. 861–877, 2008.
- [4] "Openfoam® – the open source cfd toolbox," <http://www.openfoam.com>.
- [5] E. M. Papoutsis-Kiachagias and K. C. Giannakoglou, "Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: Industrial applications," *Arch Comput Meth Eng*, 2014. DOI 10.1007/s11831-014-9141-9.
- [6] G. K. Karpouzias and E. de Villiers, "Level-set based topology optimization using the continuous adjoint method," in *OPTi 2014, Internat Conf Eng Appl Sciences, Kos, Greece*, 2014.
- [7] C. Othmer, "Adjoint methods for car aerodynamics," *J Math Ind*, vol. 4, no. 6, 2014. www.mathematicsinindustry.com/content/4/1/6.
- [8] T. Borrvall and J. Petersson, "Topology optimization of fluids in stokes flow," *Int J Num Meth Fluids*, vol. 41, no. 1, pp. 77–107, 2003.
- [9] O. Moos, F. Klimetzek, and R. Rossmann, "Bionic optimization of air-guiding systems," *SAE Technical Paper 2004-01-1377*, 2004.
- [10] E. de Villiers and C. Othmer, "Multi-objective adjoint optimization of intake port geometry," *SAE Technical Paper 2012-01-0905*, 2012.
- [11] E. Stavropoulou, M. Hojjat, and K. U. Bletzinger, "In-plane mesh regularization for parameter-free shape optimization problems," *Comp Meth Appl Mech Eng*, vol. 275, pp. 39–54, 2014.
- [12] M. Hojjat, E. Stavropoulou, and K. U. Bletzinger, "The vertex morphing method for shape optimization," *Comp Meth Appl Mech Eng*, vol. 268, pp. 494–513, 2014.
- [13] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer, "Continuous adjoint approach to the spalart-allmaras turbulence model for incompressible flows," *Computers & Fluids*, vol. 38, no. 8, pp. 1528–1538, 2009.
- [14] M. Islam, F. Decker, E. de Villiers, A. Jackson, J. Gines, T. Grahs, A. Gitt-Gehrke, and J. Comas i Font, "Application of detached-eddy simulation for automotive aerodynamics development," *SAE Technical Paper 2009-01-0333*, 2009.
- [15] "aboutflow – adjoint-based optimisation of industrial and unsteady flows," <http://aboutflow.sems.qmul.ac.uk>.
- [16] E. M. Papoutsis-Kiachagias, K. C. Giannakoglou, and C. Othmer, "Adjoint wall functions: Validation and application to vehicle aerodynamics," in *Proc 6th Europ Conf CFD, Barcelona, Spain*, 2014.
- [17] A. S. Zymaris, D. I. Papadimitriou, K. C. Giannakoglou, and C. Othmer, "Optimal location of suction or blowing jets using the continuous adjoint approach," in *Proc 5th Europ Conf CFD, Lisbon, Portugal*, 2010.
- [18] R. Petzold, "Experimentelle untersuchungen zur strömungsbeeinflussung an einem 1:4-fahrzeugmodell," *Diploma thesis, TU Braunschweig, Inst. für Strömungsmechanik*, 2011.
- [19] H. Narten, C. Correia, C. Othmer, and R. Radespiel, "Adjoint-based cooling efficiency optimization of turbulent ducted flows," in *ThermaComp 2014, 3rd Internat Conf Comput Meth Thermal Problems, Lake Bled, Slovenia*, 2014.
- [20] K. C. Giannakoglou, "The continuous adjoint method for the computation of first- and higher-order sensitivities," in *MUSAF II, Multiphysics and Unsteady Simulations for Aeronautical Flows, Toulouse, France*, 2013.

MULTI-OBJECTIVE DRAG REDUCTION DESIGN OF NATURAL LAMINAR AIRFOIL WITH SHOCK CONTROL BUMP SHAPE USING EVOLUTIONARY ALGORITHMS AND PARETO GAMES

Z. Tang¹, Y. Chen¹, J. Periaux²

¹ *Nanjing University of Aeronautics and Astronautics, Nanjing, China*

² *CIMNE, Barcelona, Spain*

Key words: natural-laminar-flow airfoil, linear stability theory, transition prediction, shock wave reduction, Bump, multi-objective optimization, Genetic algorithms, non dominated Pareto members

Abstract

In order to improve the flight performances and to increase cruise time of transonic high aspect ratio aircraft, it is important to carry out the research of friction drag reduction (mainly turbulent friction). At high Reynolds numbers, the Laminar Flow Control technology and Natural Laminar Flow profile/wing design are efficient methods which can reduce the turbulence skin friction. However, the existence of wide range of favorable pressure gradient in laminar flow airfoil surface leads to strong shock wave on the neighborhood of the trailing edge of an airfoil. Therefore, the reduction of the friction drag is competitively balanced with the increase of shock wave induced drag. In this paper, we introduce the transition prediction method based on the linear stability theory to predict the transition location of the airfoil. Then a Multi-objective Genetic Algorithm (GAs) coupled with a Nash game is implemented to optimize the airfoil shape with larger laminar flow range and weaker shock wave drag simultaneously due to a Shock Control Bump (SCB). The associated coupled software can easily capture the Pareto Front of this Multi-objective optimization problem. The non dominated solutions of the front show the aerodynamic performance airfoils trade offs between the delay of the profile's transition location and the increased intensity of shock wave due the bump installed at the upper surface of the airfoil. Results of numerical experiments demonstrate that a series of laminar flow airfoils which higher aerodynamic performance is greatly improved with the SBC compared to the baseline airfoil.

1 Introduction

Once the total drag of a civil transport aircraft or an Unmanned Aerial Vehicle(UAV) is decomposed, one can find that skin friction drag and lift-induced drag are two major sources, and are approximately forty percent and one third of the total drag respectively, see Fig. 1. Drag reduction directly impacts the Direct Operating Cost (DOC): 1% of drag reduction can lead to 0.2% of DOC decrease or in the other word 1.6 tons of operating empty weight decrease[1]. In view of this, laminarity is an efficient way to reduce drag greatly in transonic condition. NASA flight tests show that when the laminar flow area of the wing surface expands from 10% to 90%, the flying range can be increased by nearly 50%[2].

Nowadays, most of civil transport aircraft can achieve very high lift-to-drag ratio in transonic regime due to the development of supercritical airfoil. However, the unsteadiness of the cross flow in the boundary layer makes the flow transition to turbulent at high Reynolds number (e.g. 106 to 108), consequently the skin friction drag increases rapidly. The design of high aspect ratio transonic laminar aircraft has a great influence on the flight time of the UAV. This flight configuration provides the UAV a better ability in intelligence gathering and battlefield surveillance. Meanwhile, the laminar airfoil technology can also reduce the CO₂ emissions from the aircraft, improving significantly the operational efficiency [3].

Maintaining a wide favorable pressure gradient is the way to obtain a laminar airfoil. However, this laminarity of the airfoil could lead to a stronger shock wave on its trailing edge due to the recovery of the pressure, the skin friction drag decreasing but the wave drag increasing simultaneously. In this short report, the natural laminar airfoil with bump shock control (BSC) is designed in the objective of drag reduction of the airfoil at transonic cruise regime. Then a multi Multi-objective optimization method with Genetics Algorithms coupled to Pareto games is used to obtain a natural laminar airfoil which has through cooperative games a wider laminar flow area and a weaker shock wave.

2 Flow Field Analysis Coupled with Viscous Boundary Layer Correction

The e^N method based on linear stability analysis needs the velocity profile within boundary layer. So, in this section, the pressure coefficient around the airfoil is computed by solving 2-D RANS. Then the accurate velocity profile is obtained by solving the boundary layer equation based on the output of the RANS equation.

RANS equations is solved by using a finite volume Galerkin method on structured meshes. To solve the Euler part of the equations, a Roe scheme has been used. To compute turbulent flows a $k - \epsilon$ model has been chosen. Near-wall turbulence has been computed by a two-layer approach. Time-dependant problems have been solved using a fourth-order Runge-Kutta scheme.

Among numerical methods for solving the boundary layer equations, the Crank-Nicolson and Keller's box methods are the most convenient ones. Keller's method has significant advantages compared to the other and

in this section it is used to solve the boundary layer equations for 2-D flows. The governing equation and the boundary conditions are discretized through the method of Keller's box. The resulting system of equations which is implicit and nonlinear is linearized by Newton's method and solved by the block-elimination method [4,5].

3 The e^N Method for Transition Prediction

The transition prediction always is a difficult problem in hydromechanics. The Tollmien-Schlichting waves, the cross flow attached line transition and the bypass transition are the main factors which influence the flow transition from laminar to turbulence. Transition caused by the Tollmien-Schlichting waves is only considered since due to 2-D optimization in this paper.

The e^N method is based on small disturbance theory in which a small sinusoidal disturbance is imposed on a given steady laminar flow to see whether the disturbance will amplify or decay in time. If the disturbance decays, the flow will stay laminar while if the disturbance amplifies sufficiently, the flow must transit to turbulent.

4 Multi-Objective Laminar Airfoil Design Optimization with Shock Control Bump

Until the 70's, it was known that the critical Reynolds number was not the most important factor to effect transition location on airfoil. The transition starts when the disturbance amplified to a certain value. So the amplification of the disturbance in the flow was the important factor. For the appropriate design airfoil, transition may start at 0.7 chord length in the flow with favorable pressure gradient. Consequently, it is possible to design an airfoil to delay the transition to turbulence.

4.1 Multi-Objective Optimization Method

The approach described in the previous section requests one (or a set) of solution(s) of a multi-objective optimization problem. There exists several variants of GAs for multi-objective optimization problems; see for example Vector Evaluated GAs (VEGAs)[6] and Non-dominated Sorting GAs (NSGAs)[7]. For further information on

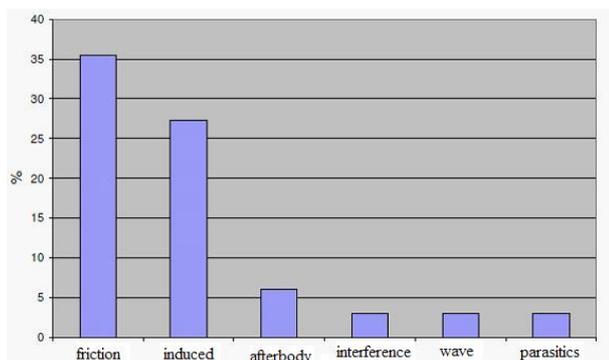


Figure 1: Drag decomposition of a typical civil transport aircraft

GAs to solve multi-objective optimization problem, the reader may consult Refs [8] [9] and references thereafter.

The test cases presented in this paper have been solved using a modified version of a Non-dominated Sorting Genetic Algorithm (NSGA). More recent algorithms such as SPEA2 [9], epsilon-MOEA[10], or NSGA-II[11] (an improved version of NSGA) could also have been chosen. Nevertheless, the main objective of this work is the integration of different tools for the solution of laminar airfoil design problems. Obviously, any improvement in any of the integrated GAs and Pareto game tools should produce a improved performance efficiency of the whole procedure.

4.2 The Numerical Optimization of the Bump Shape to Reduce Wave Drag

In order to delay the transition location, the airfoil should have a wider area which have favorable pressure gradient in its laminarity behaviour at transonic regimes. But the existence of a wide range of favorable pressure gradient in airfoil surface leads to strong shock wave on the neighborhood of the trailing edge of the airplane. Consequently, an increase of the shock wave drag is associated to the reduction of the friction. Then, a method of drag reduction, namely a control device named a bump, is used for the design of the natural laminar airfoil. Among the different active/passive shock boundary layer control concepts investigated, the bump concept proposed by P. Ashill and trailing edge devices seem very promising.

The bump concept is based on the local modification of the airfoil surface in the shock region. The straight shock is transformed into a lambda shock configuration and its strength is reduced by the presence of the compression waves. In this article, Hicks- Henne functions are used to control the bump shape optimization.

The height, length and location of the bump are design parameters which influence the strength of the shock wave. Therefore such design parameters are considered in the optimization procedure. In this section, the RAE2822 airfoil is chosen as baseline and is optimized with the objective of minimizing the shock wave.

In the Fig. 3, it shows the process of the optimization. From the figure, it can be known that the optimal solution was obtained at the 20th generation. And the drag coefficient decreased from the initial 0.042 to the 0.036. The lift and drag coefficient of the initial airfoil and the airfoil with bump are given in the Table 1. In the table, it can be known that the aerodynamic performance of the airfoil with the bump was improved. The lift coefficient increased by 3.17%, and the drag coefficient decreased by 7.32%. In the figure 4, 5 and 6, it shows that

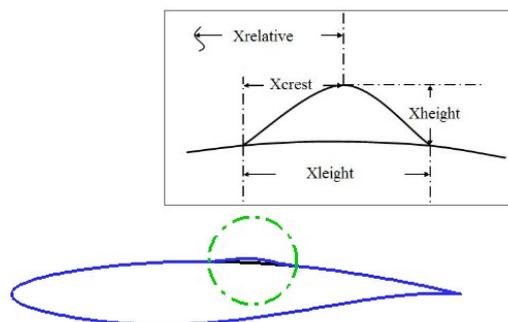


Figure 2: Shape and position of the bump

Table 1: Lift and drag coefficient of the baseline airfoil and the airfoil with bump

Aerodynamic force	RAE2822 airfoil	Airfoil with bump	$\Delta(\%)$
C_L	0.821	0.847	+3.17
C_D	0.041	0.038	-7.32
C_L/C_D	20.02	22.41	+11.94

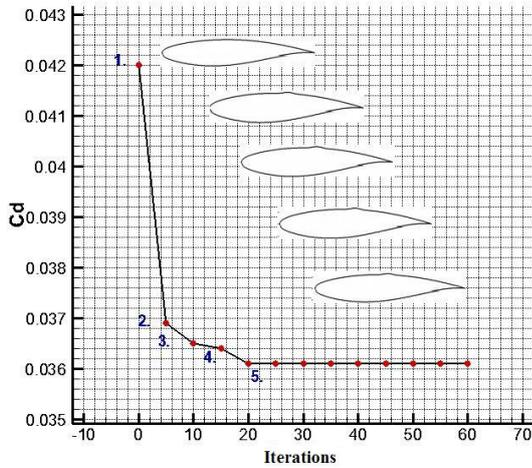


Figure 3: Convergence of the GAs optimization

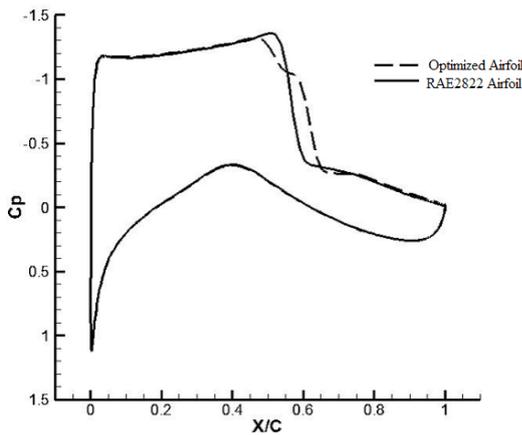


Figure 4: Pressure distributions of the baseline and optimized airfoils

the strength of the shock wave of the optimal airfoil was obvious weaker than the baseline airfoil. It turned out that the bump could reduce significantly the shock wave of the airfoil.

4.3 The Natural Laminar Airfoil Design Optimization

In this section, the RAE2822 airfoil is chosen as baseline and optimized with the objective of maximizing the range of the laminar flow area in order to verify the laminarity feasibility of the designed airfoil.

At the converged optimization, it can be known on Fig. 7 and Table 2, that the transition location of the airfoil is decayed and reaches to 57% chord for the upper surface and 52% for lower surface for the optimized airfoil. However, it can also be known that the drag force increased. That is to say, the drag coefficient increased with the increase of the laminar area.

Fig. 5 and Fig.8 show that the strength of the shock wave of the optimized airfoil is stronger than the baseline airfoil.

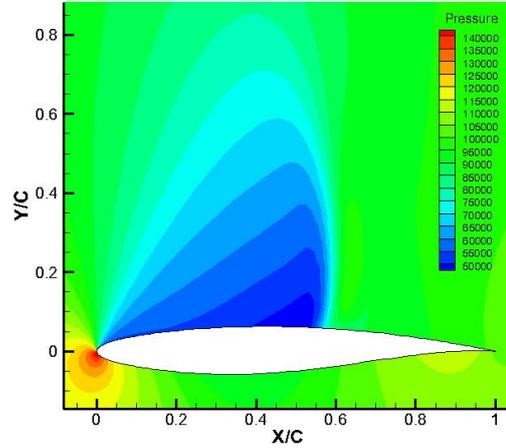


Figure 5: Pressure contours of the RAE2822 airfoil

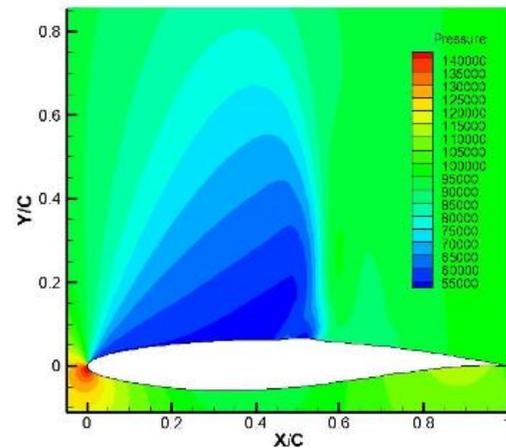


Figure 6: Pressure contours of the airfoil with bump

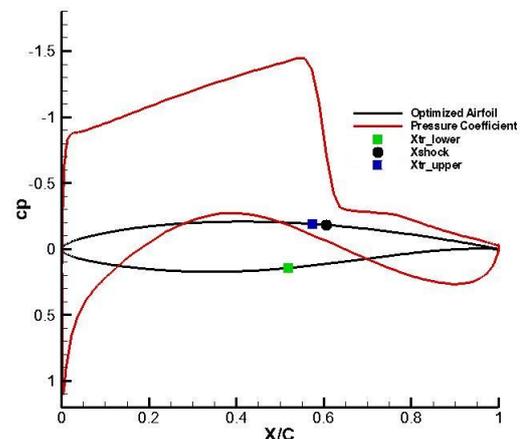


Figure 7: Pressure distribution of the optimal airfoil

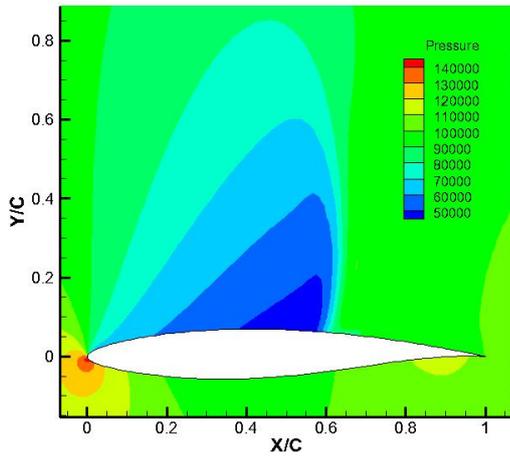


Figure 8: Pressure contours of the optimal airfoil

Table 2: Lift coefficient, drag coefficient and transition location of the baseline and optimal airfoil

	RAE2822 airfoil	Optimized airfoil
C_L	0.821	0.823
C_D	0.041	0.048
X_{tr_upper}/C	1.36	0.537
X_{tr_lower}/C	0.472	0.521

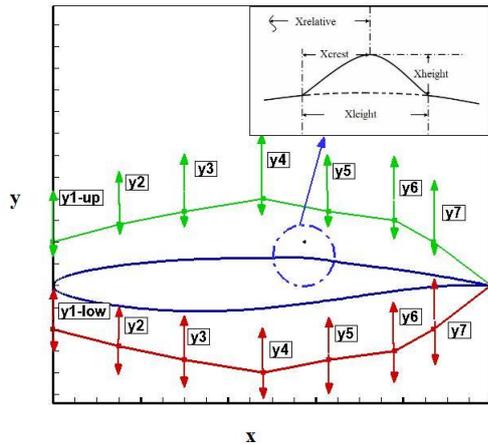


Figure 9: The parameters distribution of the airfoil and bump

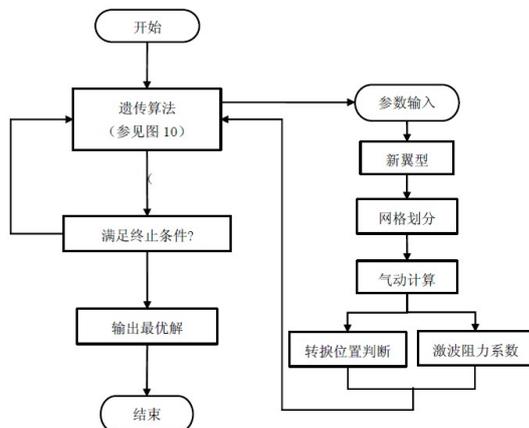


Figure 10: The flow chart of the optimization

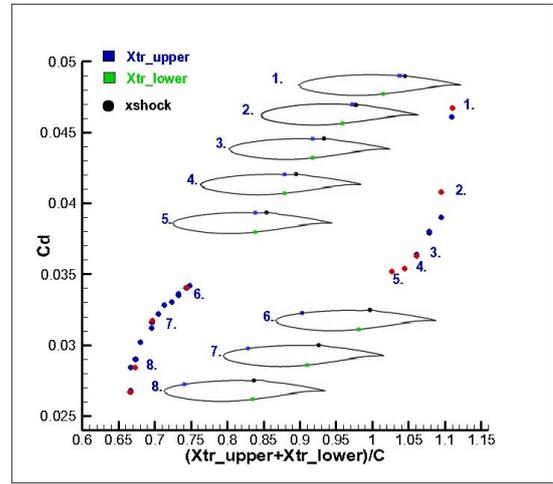


Figure 11: Converged Pareto Front

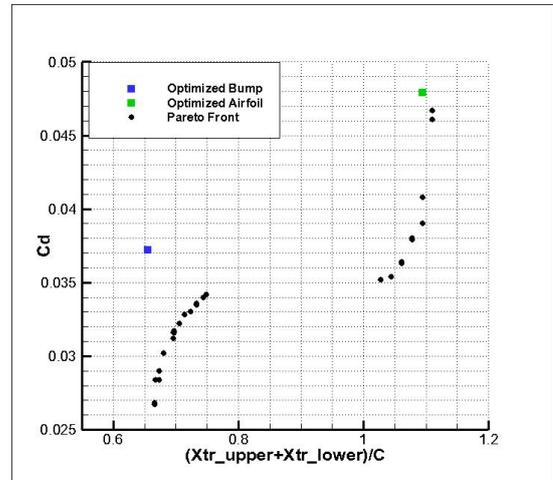


Figure 12: Transition location and drag coefficient of the airfoil with optimal bump, laminar airfoil and laminar airfoil with bump

In this article, in order to design the natural laminar airfoil with optimal bump shape and location, the two following optimization problems are considered :

1. Maximize the range of the laminar flow;
2. Decrease the strength of the shock wave of the airfoil.

The objective functions are defined as follows :

$$\left\{ \begin{array}{l} \max (X_{tr_upper} + X_{tr_lower}) + 3 \times \beta \\ \quad \times (Cl - Cl_0) / Cl_0 \times (X_{tr_upper} + X_{tr_lower}) \\ \min 30 \times (Cd_w - 3 \times \beta \times (Cl - Cl_0) / Cl_0 \times Cd_w) \\ \text{subject to:} \\ Cl \geq Cl_0 \quad \beta = 0 \\ Cl \leq Cl_0 \quad \beta = 1 \end{array} \right.$$

where X_{tr_upper} and X_{tr_lower} represent transition location of the upper and lower surface of the airfoil respectively and Cd_w represents the shock wave coefficient. Then, the RAE2822 airfoil is chosen as baseline and optimized. On Fig. 9 with Tables 3 and 4, are shown design parameters and the value upper/lower range. Fig.10 provides the flow chart of the optimization.

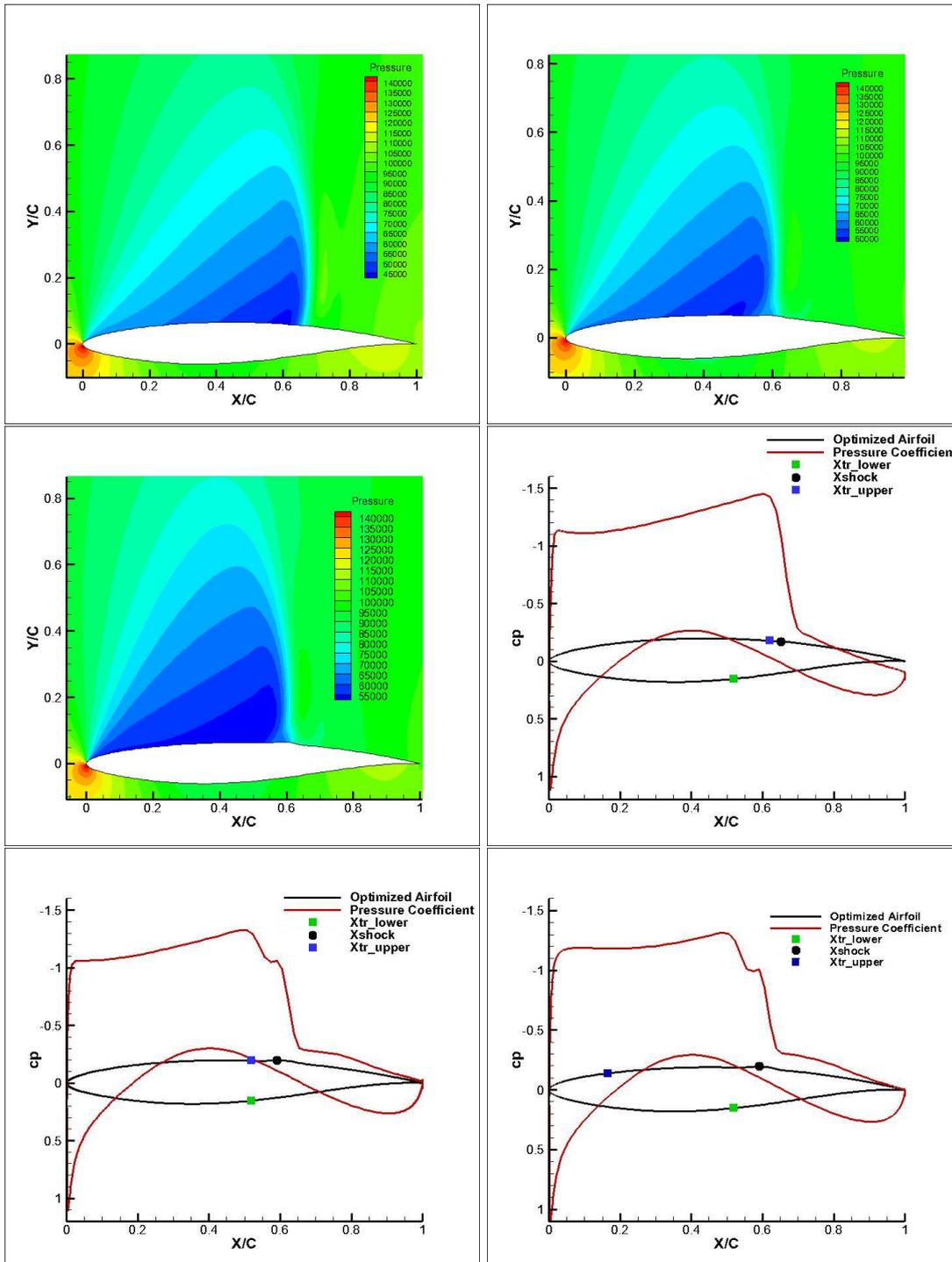


Figure 13: Pressure distribution, pressure contours and transition location of the PM2, PM3 and PM4 of Table 4

Table 3: Upper/lower Range of the baseline airfoil design parameters

airfoil	Upper surface		Lower surface	
	Upper	Lower	Upper	Lower
y1	-0.005	0.002	-0.003	0.002
y2	-0.008	0.003	-0.008	0.003
y3	-0.008	0.015	-0.008	0.005
y4	-0.005	0.025	-0.008	0.008
y5	-0.005	0.015	-0.005	0.005
y6	-0.005	0.005	-0.005	0.003
y7	-0.003	0.003	-0.001	0.001

Table 4: Upper/lower range of the bump design parameters

bump	X relative/C	X length/C	X height/C	X crest/C
maximization	0.65	0.25	0.007	0.5, 0.55
minimization	0.5	0.1	0.001	0.6, 0.65

Table 5: The aerodynamic force and transition location of the airfoil

		C_L	C_D	X_{tr_upper}	X_{tr_lower}	X_{shock}
RAE2822 airfoil		0.821	0.0413	0.136	0.472	0.553
Bump optimized only		0.847	0.0375	0.151	0.504	0.551
Airfoil optimize only		0.823	0.0479	0.573	0.521	0.606
Laminar airfoil with shock control	1	0.823	0.0468	0.622	0.521	0.654
	2	0.814	0.0413	0.589	0.520	0.606
	3	0.822	0.0365	0.523	0.521	0.590
	4	0.807	0.0357	0.521	0.521	0.590
	5	0.801	0.0354	0.503	0.521	0.590
	6	0.834	0.0341	0.162	0.521	0.590
	7	0.832	0.0336	0.152	0.521	0.590
	8	0.820	0.0322	0.127	0.521	0.590

4.4 The Design of the Natural Laminar Airfoil with Shock Control Bump

On Fig. 11, the Pareto front of the Multi objective optimization problem is presented with its non dominated solutions describing tradeoffs between the increased shock wave intensity and the increased range of the laminar flow. The strength of the airfoil shock wave which has the widest range of laminar flow is evidently stronger than the one of the baseline. Then it can be observed that Pareto members (PMs) performances of both the laminar flow area and the shock wave intensity of the airfoil located on the middle of the Pareto Front are improved when compared to baseline drag performances. From the figure 12, it can be observed that the strength of the shock wave decreases with optimizing the bump only, but the transition location still at the leading edge of the airfoil, with a skin friction drag still very large. At another condition with optimized airfoil only, the transition location was delayed to 50% of the chord compared with 10% chord of the baseline airfoil, but the strength of the induced shock wave was stronger. On Table 4, it is shown the aerodynamic force of the eight (8) non dominated airfoils on the Pareto front. Fig. 23 presents pressure contours, pressure distribution and transition location of Pareto members airfoils PM2, PM3 and PM4 with their performances shown of Table 4.

5 Conclusion

In this paper, the RANS equations coupled with the boundary layer equations are solved to obtain the velocity distribution, displacement thickness and momentum thickness of the boundary layer. Then, the result of the boundary layer is used to solve the Orr-Sommerfeld equation, and obtain the spatial disturbance amplification. NLF416 and NACA0012 airfoils are chosen as baseline to verify the accuracy of the method. It turned out that the method has a high accuracy and can be used to predict the transition location of the airfoil.

However, the existence of wide range of favorable pressure gradient in laminar flow airfoil surface ignites a strong shock wave on the neighborhood of the trailing edge of aircraft. Consequently, the reduction of the friction drag generates an increase of the shock wave resistance. In this paper, the shock control method was used to decrease the strength of the wave drag.

The design of the natural laminar airfoil with a shock control bump is the main problem. After a research investigation on laminar airfoils and shock controls devices, the RAE2822 airfoil is chosen as baseline and studied with Genetic Algorithms (GAs) and Pareto games with the two objectives of maximizing the range of the laminar flow and minimizing the strength of the shock wave.

After optimization, the Pareto front associated to the two objectives design problem is captured. The non dominated solutions on the front show aerodynamic performance tradeoffs of different profiles/bumps geometries. On the Pareto front different delays of the profile's transition location associated with different intensities of shock waves on the upper surface are optimally enhanced. It is proposed to the designer a set of laminar flow airfoils obtained numerically with aerodynamic performances significantly improved when compared to the baseline airfoil.

References

- [1] Reneaux J. Overview On Drag Reduction Technologies for Civil Transport aircraft, ECCOMAS 2004.
- [2] Ronald D. Joslin. Overview of Laminar Flow Control, NASA/TP-1998-208705.
- [3] John E. Green. Laminar Flow Control-Back to the Future?. 38th Fluid Dynamics Conference and Exhibit, 23-26 June 2008.
- [4] Cebeci, T. An Engineering Approach to the Calculation of Aerodynamic Flows, Heideberg, 1999.
- [5] Tuncer Cebeci, Jean Cousteix. Modeling and Computation of Boundary-Layer Flows, Horizons Publishing 2003.
- [6] Jen-Der Lee, Antony Jameson. Natural-Laminar-Flow Airfoil and Wing Design by Adjoint Method and Automatic Transition Prediction. 47th AIAA, 5-8 January 2009.
- [7] J. Driver, D. W. Zingg. Optimized Natural-Laminar-Flow Airfoils. AIAA Journal, Vol. 40, No. 6, 2002.
- [8] Patzold, M., Lutz, T., Kramer, E., and Wagner, S., "Numerical Optimization of Finite Shock Control Bumps," AIAA 2006-1054, 2006.
- [9] H. Ruifeng. The principle and application of the Genetic Algorithm. Bei Jing: Weapons industry press, 2009.
- [10] Fletcher Hartshorn. Computational Optimization of a Natural Laminar Flow Experimental Wing Glove. 50th AIAA, 09-12 January 2012.
- [11] Micheal J. Belisle, Matthew W. Roberts. A Transonic Laminar-Flow Wing Glove Flight Experiment: Overview and Design Optimization. 30th AIAA Applied Aerodynamics Conference 25-28 June 2012.

ERCOFTAC Workshops and Summer Schools

Title	Location	Date	Coordinators	Organiser
European Drag Reduction and Flow Control	Cambridge, UK	22-27/03/2015	Choi, K-S., Garcia-Mayoral, R	SIG20
Ercoftac Montestigliano Spring School - A Workshop Series in Fluid Dynamics for Doctoral Students	Montestigliano, Italy	11-18/04/2015	Bagheri, Sh.,	ERCOFTAC Core Activity
11th ERCOFTAC Workshop – Progress in Transition Modelling and Control	Jersey, UK	15-17/04/2015	Hanifi, A	SIG33
Understanding and Harnessing Turbulence Honouring W.K George on Occasion of His 70th Birthday	Sain Etienne du Rouvray, France	20-24/04/2015	Danaila, L.	PC France West
A Systems Approach to Turbulence with Scalar Transport, Buoyancy Effects and Possible Confinement	Paris, France	05-06/05/2015	Gomez, T., Cambon, C.	SIG35, SIG24, PC Henri Benard, PC France West
Turbulent Combustion of Sprays Workshop	Rhodes, Greece	07/06/2015	Mastorakos, E. Merci, B.	SIG28
Synthetic Flow Structure – Clustering and Modelling	Lyon, France	29-30/06/2015	Nicolleau, F.	SIG42, SIG35
14th Workshop on Two – Phase Flow Predictions	Halle, Germany	07-10/09/2015	Sommerfeld, M.	SIG12

INDUSTRY EVENTS 2015
ORGANISER: DR. RICHARD E. SEOUD

Oil & Gas: Modelling & Simulation – Best Practice & Technology Trends

9-10 March 2015, TNO, Delft, Netherlands

Coordinator: Dr Richard E Seoud

Transition Modelling

21-22 May 2015, GE, Munich, Germany

Coordinator: Dr Andreas Krumbein, DLR, Germany

Design Optimization: Methods & Applications

28-29 May 2015, FOI, Stockholm, Sweden

Coordinator: Prof. Kyriakos Giannakoglou, NTUA, Greece

Hybrid RANS-LES Methods for Industrial CFD: Overview, Guidance, & Examples

September 2015, EU

Coordinator: Prof. Shia-Hui Peng, FOI, Stockholm

CFD for Dispersed Multiphase Flows

1-2 October 2015, Imperial College, London, UK

Coordinator: Prof. Martin Sommerfeld, Univ. Halle-Wittenberg, Germany

Lattice Boltzmann Methods for Industrial Applications: Overview, Guidance & Examples

19-20 October 2015, Aix-Marseille Universite, Marseille, France

Coordinator: Prof. Pierre Sagaut, Aix-Marseille Univeriste, France

Best Practice for Eng CFD

22-23 October 2015, GE, Munich, Germany

Coordinator: Prof. Charles Hirsch, Em. Vrije Universiteit Brussel, Belgium & President, Numeca International

Best Practices in Combustion CFD, Flame Stabilization, and Combustion Instabilities

3-4 November 2015, TU Delft, Netherlands

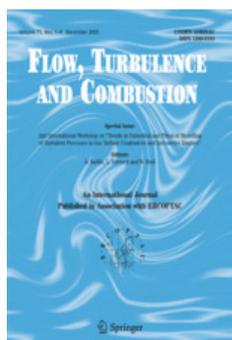
Coordinators: Prof. Luc Vervisch, Coria, Rouen, France & Prof. Dirk Roekaerts, TU Delft

Uncertainty Management & Quantification in Industrial Design & Analysis

Autumn 2015

Coordinator: Prof. Charles Hirsch, Em. Vrije Universiteit Brussel, Belgium & President, Numeca International

Industry Events 2015
Enquiries & Registration: richard.seoud-ieo@ercoftac.org



RECENT TABLE OF CONTENTS OF FLOW TURBULENCE AND COMBUSTION

AN INTERNATIONAL JOURNAL PUBLISHED BY SPRINGER
IN ASSOCIATION WITH ERCOFTAC

EDITOR-IN-CHIEF: K. HANJALIĆ
EDITORS: J.J. CHEN, M. REEKS, W. RODI, L. VERVISCH
HONORARY EDITOR: J.C.R HUNT
FOUNDING EDITOR: F. NIEUWSTADT

VOLUME 94, NUMBER 1, JANUARY 2015

Special Issue: Turbulence and Shear Flow Phenomena

Guest Editors: Jean-Paul Bonnet, Hyung Jin Sung, Luc Vervisch, Kemo Hanjalić

Preface

J.-P. Bonnet, H.J. Sung, L. Vervisch, K. Hanjalić

Modeling Combustion Chemistry in Large Eddy Simulation of Turbulent Flames

B. Fiorina, D. Veynante, S. Candel

Feedback Control for Laminarization of flow over Wings

R. Dadfar, A. Hanifi, D.S. Henningson

Extension of Lysaks Approach to Evaluate the Wall Pressure Spectrum for Boundary Layer Flows

B. Aupoix

An Experimental Investigation of Turbulent Convection Velocities in a Turbulent Boundary Layer

C. Atkinson, N.A. Buchmann, J. Soria

Active Flow Control for High Speed Jets with Large Window PIV

Z.P. Berger, P.R. Shea, M.G. Berry, B.R. Noack, S. Gogineni, M.N. Glauser

Roughness Effects on Turbulent Flow Downstream of a Backward Facing Step

E.E. Essel, M.F. Tachie

Mixing Layer Manipulation Experiment

V. Parezanović, J.-C. Laurentie, C. Fourment, J. Delville, J.-P. Bonnet, A. Spohn, T. Duriez, L. Cordier, B.R. Noack, M. Abel, M. Segond, T. Shaqarin, S.L. Brunton

LES Investigation of the Hysteresis Regime in the Cold Model of a Rotating-Pipe Swirl Burner

R. Mullyadzhanov, M. HadÅiabdi, K. Hanjalić

Flame Wrinkling Factor Dynamic Modeling for Large Eddy Simulations of Turbulent Premixed Combustion

T. Schmitt, M. Boileau, D. Veynante

Quantification of the Pre-ignition Front Propagation in DNS of Rapidly Compressed Mixture

G. Lodier, P. Domingo, L. Vervisch

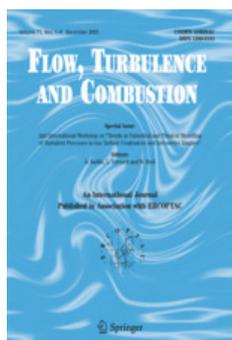
Flame Dynamics During Combustion Instability in a High-Pressure, Shear-Coaxial Injector Combustor

S. Srinivasan, R. Ranjan, S. Menon

Multi-Scale High Intensity Turbulence Generator Applied to a High Pressure Turbulent Burner

R. Fragner, N. Mazellier, F. Halter, C. Chauveau, I. Gökalp

Further articles can be found at www.springerlink.com



RECENT TABLE OF CONTENTS OF FLOW TURBULENCE AND COMBUSTION

AN INTERNATIONAL JOURNAL PUBLISHED BY SPRINGER
IN ASSOCIATION WITH ERCOFTAC

EDITOR-IN-CHIEF: K. HANJALIĆ

EDITORS: J.J. CHEN, M. REEKS, W. RODI, L. VERVISCH

HONORARY EDITOR: J.C.R HUNT

FOUNDING EDITOR: F. NIEUWSTADT

VOLUME 94, NUMBER 2, MARCH 2015

An Investigation into In-Cylinder Tumble Flow Characteristics with Variable Valve Lift in a Gasoline Engine

T. Wang, D. Liu, B. Tan, G. Wang, Z. Peng

Pulsatile Turbulent Flow in Straight and Curved Pipes Interpretation and Decomposition of Hot-Wire Signals

A. Kalpakli Vester, R. Örlü, P.H. Alfredsson

Investigation of Glycerol Atomization in the Near-Field of a Flow-Blurring Injector using Time-Resolved PIV and High-Speed Visualization

L. Jiang, A.K. Agrawal

Study of the Discrete Spectrum in a Mach 4.5 Görtler Flow

J. Ren, S. Fu

Curvature Effects in Turbulent Premixed Flames of H_2 /Air: a DNS Study with Reduced Chemistry

G. Rocco, F. Battista, F. Picano, G. Troiani, C.M. Casciola

RANS Simulations of Statistically Stationary Premixed Turbulent Combustion Using Flame Speed Closure Model

E. Yasari, S. Verma, A.N. Lipatnikov

A Numerical Simulation of the Effects of Swirling Flow on Jet Penetration in a Rotating Channel

N. Al-Zurfi, A. Turan

Assessment of Turbulence-Chemistry Interaction Models in MILD Combustion Regime

A. De, A. Dongre

Further articles can be found at www.springerlink.com

Abstracted/Indexed in Science Citation Index, Science Citation Index Expanded (SciSearch), Journal Citation Reports/Science Edition, SCOPUS, INSPEC, Chemical Abstracts Service (CAS), Google Scholar, EBSCO, CSA, ProQuest, Academic OneFile, ASFA, Current Abstracts, Current Contents/Engineering, Computing and Technology, Earthquake Engineering Abstracts, EI Encompass, Ei Page One, EI-Compendex, EnCompassLit, Engineered Materials Abstracts, Gale, OCLC, PASCAL, SCImago, STMA-Z, Summon by Serial Solutions, VINITI - Russian Academy of Science.

Instructions for Authors for Flow Turbulence Combust are available at <http://www.springer.com/10494>.

ERCOFTAC Special Interest Groups

1. Large Eddy Simulation

Salvetti, M.V.
University of Pisa, Italy.
Tel: +39 050 221 7262
Fax: +39 050 221 7244
mv.salvetti@ing.unipi.it

4. Turbulence in Compressible Flows

Dussauge, Jean-Paul
IUSTI, Marseille
jean-paul.dussauge
@polytech.univmrs.fr

5. Environmental Fluid Mechanics

Armenio, V.
Universit di Trieste, Italy.
Tel: +39 040 558 3472
Fax: +39 040 572 082
armenio@dica.units.it

10. Transition Modelling

Dick, E.,
University of Ghent, Belgium.
Tel: +32 926 433 01
Fax: +32 926 435 86
erik.dick@ugent.be

12. Dispersed Turbulent Two Phase Flows

Sommerfeld, M.
Martin-Luther University, Germany.
Tel: +49 346 146 2879
Fax: +49 346 146 2878
martin.sommerfeld@iw.uni-halle.de

14. Stably Stratified and Rotating Flows

Redondo, J.M.
UPC, Spain.
Tel: +34 934 017 984
Fax: +34 934 016 090
redondo@fa.upc.edu

15. Turbulence Modelling

Jakirlic, S.
Darmstadt University of Technology,
Germany.
Tel: +49 615 116 3554
Fax: +49 615 116 4754
s.jakirlic@sla.tu-darmstadt.de

20. Drag Reduction and Flow Control

Choi, K-S.
University of Nottingham, England.
Tel: +44 115 951 3792
Fax: +44 115 951 3800
kwing-so.choi@nottingham.ac.uk

24. Variable Density Turbulent Flows

Anselmet, F.
IMST, France.
Tel: +33 491 505 439
Fax: +33 491 081 637
anselmet@irphe.univ-mrs.fr

28. Reactive Flows

Roekaerts, D.
Delft University of Technology,
The Netherlands.
Tel: +31 152 782 470
D.J.E.M.Roekaerts@tudelft.nl

32. Particle Image Velocimetry

Stanislas, M.
Ecole Centrale de Lille, France.
Tel: +33 320 337 170
Fax: +33 320 337 169
Michel.Stanislas@ec-lille.fr

33. Transition Mechanisms, Prediction and Control

Hanifi, A.
FOI, Sweden.
Tel: +46 855 503 197
Fax: +46 855 503 397
ardeshir.hanifi@foi.se

34. Design Optimisation

Giannakoglou, K.
NTUA, Greece.
Tel: +30 210 772 1636
Fax: +30 210 772 3789
kgianna@central.ntua.gr

35. Multipoint Turbulence Structure and Modelling

Cambon, C.
ECL Ecully, France.
Tel: +33 472 186 161
Fax: +33 478 647 145
claude.cambon@ec-lyon.fr

36. Swirling Flows

Braza, M.
IMFT, France.
Tel: +33 534 322 839
Fax: +33 534 322 992
braza@imft.fr

37. Bio-Fluid Mechanics

Poelma, C.
Delft University of Technology, Holland.
Tel: +31 152 782 620
Fax: +31 152 782 947
c.poelma@tudelft.nl

38. Micro-thermofluidics

Borhani, N.
EPFL, Switzerland.
Tel: +41 216 933 503
Fax: +41 216 935 960
navid.borhani@epfl.ch

39. Aeroacoustics

Bailly, C.
Ecole Centrale de Lyon, France.
Tel: +33 472 186 014
Fax: +33 472 189 143
christophe.bailly@ec-lyon.fr

40. Smoothed Particle Hydrodynamics

Le Touze, D.
Ecole Centrale de Nantes, France
Tel: +33 240 371 512
Fax: +33 240 372 523
David.LeTouze@ec-nantes.fr

41. Fluid Structure Interaction

Longatte, E.
EDF, France.
Tel: +33 130 878 087
Fax: +33 130 877 727
elisabeth.longatte@edf.fr

42. Synthetic Models in Turbulence

Nicolleau, F.
University of Sheffield, England.
Tel: +44 114 222 7867
Fax: +44 114 222 7890
f.nicolleau@sheffield.ac.uk

43. Fibre Suspension Flows

Lundell, F.
The Royal Institute of Technology,
Sweden.
Tel: +46 87 906 875
fredrik@mech.kth.se

44. Fundamentals and Applications of Fractal Turbulence

Fortune, V.
Université Pierre et Marie Curie, France.
Tel: +33 549 454 044
Fax: +33 549 453 663
veronique.fortune@lea.univ-poitiers.fr

45. Uncertainty Quantification in Industrial Analysis and Design

Lucor, D.
d'Alembert Institute, France.
Tel: +33 (0) 144 275 472
didier.lucor@upmc.fr

ERCOFTAC Pilot Centres

Alpe - Danube - Adria

Steiner, H.
Inst. Strömungslehre and
Wärmeübertragung
TU Graz, Austria
kristof@ara.bme.hu

Belgium

Geuzaine, P.
Cenaero,
CFD Multi-physics Group,
Rue des Frères Wright 29,
B-6041 Gosselies,
Belgium.
Tel: +32 71 919 334
philippe.geuzaine@cenaero.be

Brasil

Rodriguez, O.
Department of Mechanical Engineering,
Sao Carlos School of Mechanical
Engineering,
Universidade de Sao Paulo,
Brasil.
oscarmhr@sc.usp.br

Czech Republic

Bodnar, T.
Institute of Thermomechanics AS CR,
5 Dolejskova,
CZ-18200 Praha 8,
Czech Republic.
Tel: +420 224 357 548
Fax: +420 224 920 677
bodnar@marian.fsik.cvut.cz

Spain

Onate, E.
Universitat Politècnica de Catalunya,
Theofilis, V.
Universidad Politécnica de Madrid, Spain
Spain.
onate@cimne.upc.edu
vassilis@aero.upm.es

France South

Braza, M.
IMF Toulouse,
CNRS UMR - 5502,
Allée du Prof. Camille Soula 1,
F-31400 Toulouse Cedex, France.
Tel: +33 534 322 839
Fax: +33 534 322 992
Braza@imft.fr

France West

Danaila, L.
CORIA, University of Rouen,
Avenue de l'Université BP12,
76801 Saint Etienne du Rouvray
France.
Tel: +33 232 953 702
luminita.danaila@coria.fr

Germany North

Gauger, N.R.
Chair for Scientific Computing
TU Kaiserslautern
Paul-Ehrlich-Strasse 34
67663 Kaiserslautern, Germany
Tel: +49 631 205 5635
Fax: +49 631 205 3056
nicolas.gauger@sci.comp.uni-kl.de

Germany South

Becker, S.
Universität Erlangen, IPAT
Cauerstr. 4
91058 Erlangen
Germany
Tel: +49 9131 85 29451
Fax: +49 9131 85 29449
sb@ipat.uni-erlangen.de

Greece

M. Founti.
National Tech. University Of Athens,
School of Mechanical Engineering,
Lab. of Steam Boilers and
Thermal Plants,
Heroon Polytechniou 9,
15780 Zografou, Athens, Greece
Tel: +30 210 772 3605
Fax: +30 210 772 3663
mfou@central.ntua.gr

Switzerland

Jenny, P.
ETH Zürich,
Institute of Fluid Dynamics,
Sonneggstrasse 3,
8092 Zürich, Switzerland.
Tel: +41 44 632 6987
jenny@ifd.mavt.ethz.ch

Italy

Rispoli, F.
Tel: +39 064 458 5233
franco.rispoli@uniroma1.it
Borello, D.
Tel: +39 064 458 5263
domenico.borello@uniroma1.it
Sapienza University of Rome,
Via Eudossiana, 18
00184 Roma, Italy

Netherlands

Van Heijst, G.J.
J.M. Burgerscentrum,
National Research School for Fluid
Mechanics, Mekelweg 2,
NL-2628 CD Delft, Netherlands.
Tel: +31 15 278 1176
Fax: +31 15 278 2979
g.j.f.vanheijst@tudelft.nl

Nordic

Wallin, S.
Swedish Defence Research Agency FOI,
Information and Aeronautical Systems,
S-16490 Stockholm,
Sweden.
Tel: +46 8 5550 3184
Fax: +46 8 5550 3062
stefan.wallin@foi.se

Poland

Rokicki, J.
Warsaw University of Technology,
Inst. of Aero. & App. Mechanics,
ul. Nowowiejska 24,
PL-00665 Warsaw, Poland.
Tel: +48 22 234 7444
Fax: +48 22 622 0901
jack@meil.pw.edu.pl

France - Henri Bénard

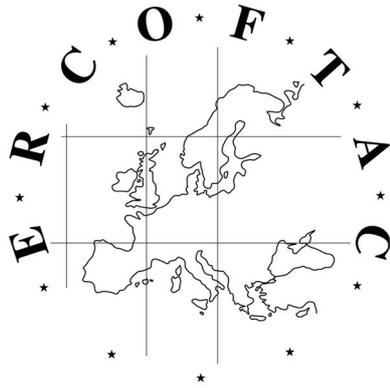
Godefert, F.S.
Ecole Centrale de Lyon.
Fluid Mechanics and Acoustics Lab.,
F-69134 Ecully Cedex,
France.
Tel: +33 4 72 18 6155
Fax: +33 4 78 64 7145
fabien.godefert@ec-lyon.fr

Portugal

da Silva, C. B.
Instituto Superior Técnico,
University of Lisbon
Av. Rovisco Pais, 1049-001 Lisboa
Portugal
carlos.silva@ist.utl.pt

United Kingdom

Standingford, D.
Zenotech Ltd.
University Gate East, Park Row,
Bristol, BS1 5UB
England.
Tel: +44 117 302 8251
Fax: +44 117 302 8007
david.standingford@zenotech.com



Best Practice Guidelines for Computational Fluid Dynamics of Dispersed Multi-Phase Flows

Editors

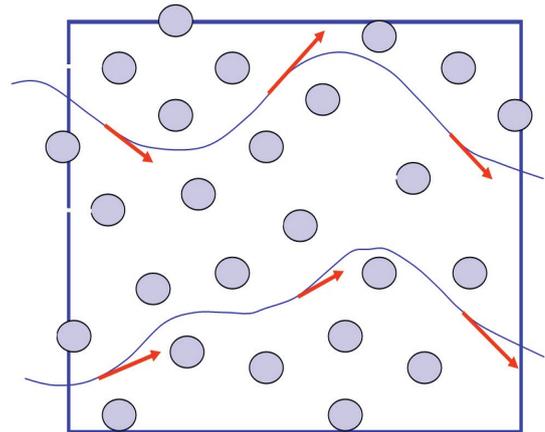
Martin Sommerfeld, Berend van Wachem
&
René Oliemans

The simultaneous presence of several different phases in external or internal flows such as gas, liquid and solid is found in daily life, environment and numerous industrial processes. These types of flows are termed multiphase flows, which may exist in different forms depending on the phase distribution. Examples are gas-liquid transportation, crude oil recovery, circulating fluidized beds, sediment transport in rivers, pollutant transport in the atmosphere, cloud formation, fuel injection in engines, bubble column reactors and spray driers for food processing, to name only a few. As a result of the interaction between the different phases such flows are rather complicated and very difficult to describe theoretically. For the design and optimisation of such multiphase systems a detailed understanding of the interfacial transport phenomena is essential. For single-phase flows Computational Fluid Dynamics (CFD) has already a long history and it is nowadays standard in the development of air-planes and cars using different commercially available CFD-tools.

Due to the complex physics involved in multiphase flow the application of CFD in this area is rather young. These guidelines give a survey of the different methods being used for the numerical calculation of turbulent dispersed multiphase flows. The Best Practice Guideline (BPG) on Computational Dispersed Multiphase Flows is a follow-up of the previous ERCOFTAC BPG for Industrial CFD and should be used in combination with it. The potential users are researchers and engineers involved in projects requiring CFD of (wall-bounded) turbulent dispersed multiphase flows with bubbles, drops or particles.

Table of Contents

1. Introduction
2. Fundamentals
3. Forces acting on particles, droplets and bubbles
4. Computational multiphase fluid dynamics of dispersed flows
5. Specific phenomena and modelling approaches
6. Sources of errors
7. Industrial examples for multiphase flows
8. Checklist of 'Best Practice Advice'
9. Suggestions for future developments



Copies of the Best Practice Guidelines can be acquired electronically from the ERCOFTAC website:

www.ercoftac.org

Or from:

ERCOFTAC (CADO)
PO Box 53877
London, SE27 7BR
United Kingdom

Tel: +44 203 602 8984

Email: magdalena.jakubczak@ercoftac.org

The price per copy (not including postage) is:

ERCOFTAC members

First copy	Free
Subsequent copies	75 Euros
Students	75 Euros

Non-ERCOFTAC academics	140 Euros
Non-ERCOFTAC industrial	230 Euros
EU/Non EU postage fee	10/17 Euros