

IT Licentiate theses
2003-012

Adjoint-based aerodynamic shape optimization

OLIVIER AMOIGNON

UPPSALA UNIVERSITY
Department of Information Technology





UPPSALA
UNIVERSITET

Adjoint-based aerodynamic shape optimization

BY
OLIVIER AMOIGNON

October 2003

DIVISION OF SCIENTIFIC COMPUTING
DEPARTMENT OF INFORMATION TECHNOLOGY
UPPSALA UNIVERSITY
UPPSALA
SWEDEN

Dissertation for the degree of Licentiate of Technology in Numerical Analysis
at Uppsala University 2003

Adjoint-based aerodynamic shape optimization

Olivier Amoignon

`Olivier.Amoignon@it.uu.se`

*Division of Scientific Computing
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden*

`http://www.it.uu.se/`

© Olivier Amoignon 2003

ISSN 1404-5117

Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

An adjoint system of the Euler equations of gas dynamics is derived in order to solve aerodynamic shape optimization problems with gradient-based methods. The derivation is based on the fully discrete flow model and involves differentiation and transposition of the system of equations obtained by an unstructured and node-centered finite-volume discretization. Solving the adjoint equations allows an efficient calculation of gradients, also when the subject of optimization is described by hundreds or thousands of design parameters.

Such a fine geometry description may cause wavy or otherwise irregular designs during the optimization process. Using the one-to-one mapping defined by a Poisson problem is a known technique that produces smooth design updates while keeping a fine resolution of the geometry. This technique is extended here to combine the smoothing effect with constraints on the geometry, by defining the design updates as solutions of a quadratic programming problem associated with the Poisson problem.

These methods are applied to airfoil shape optimization for reduction of the wave drag, that is, the drag caused by gas dynamic effects that occur close to the speed of sound. A second application concerns airfoil design optimization to delay the laminar-to-turbulent transition point in the boundary layer in order to reduce the drag. The latter application has been performed by the author with collaborators, also using gradient-based optimization. Here, the growth of convectively unstable disturbances are modeled by successively solving the Euler equations, the boundary layer equations, and the parabolized stability equations.

Contents

1	Introduction	1
2	The optimization problem	5
2.1	Shape optimization	5
2.2	Discrete adjoint in shape optimization	7
3	State equation and discretization	9
3.1	Euler equations	9
3.2	Finite-volume approximation	10
3.3	Edge-based residual assembly	12
4	Adjoint equation and reduced gradient	15
4.1	Discrete sensitivities	15
4.2	Edge-based residual assembly	17
4.3	Reduced gradient	18
5	Mesh movement and shape parameterizations	19
5.1	Mesh displacements	19
5.2	Unconstrained parameterization	20
5.3	Constrained parameterization	21
6	Applications	23
6.1	Wave drag optimization	24
6.2	Multi-point wave drag optimization	26
6.3	Delay of transition optimization	30
7	Summary and Outlook	35
	Acknowledgements	37
	Bibliography	39
A	Median dual grid	43
A.1	2D dual grid	44

A.2	3D dual grid	45
A.3	Gradients on the primal grid	45
A.3.1	Gradients ∇f_X in 2D	47
A.3.2	Gradients ∇f_X in 3D	48
B	Discrete sensitivities	51
B.1	Inner product and differentiation	51
B.2	Sensitivity calculations	55
B.2.1	Adjoint operator	57
B.2.2	Adjoint equation and reduced gradient ∇f_n	64
B.3	Detailed expressions	66
B.3.1	Adjoint boundary conditions at farfield	66
B.3.2	Adjoint artificial viscosity	66
B.3.3	Artificial viscosity in gradient expression	68
Paper A		71

Chapter 1

Introduction

The motivation of this project is to solve large-scale shape optimization problems that arise in the aeronautics industry when searching for the shape that provides the best aerodynamic performance. The approach combines gradient-based algorithms for the optimization and adjoint flow equations for the calculation of the gradients.

In the field of optimal design based on flow analysis, the use of adjoint equations can be viewed as an off-spring of the theory of optimal control for distributed parameter systems pioneered by J.L. Lions [25]. Adjoint codes for aerodynamic optimization have been developed by a number of research groups during the last decade [1, 3, 6, 8, 11, 13, 15, 33, 26, 34, 35]. In the early years following Lions work on optimal control, results were obtained by Pironneau [30] on optimal design in Stokes flows. In 1988, Jameson [23] formulated the adjoint equations for inverse optimization problems governed by the full potential and by the Euler equations.

The delay, until the first industrial applications appeared, may be due to the difficulty to formulate the adjoint equations or to implement the adjoint code [21]. The interest for adjoint codes is growing because of the efficiency of the method when applied to large-scale shape optimization problems.

Two numerical approaches compete when applying the theory of optimal control to aerodynamic shape optimization problems. In the so-called continuous approach, the adjoint problem and the expressions of the gradients are derived from the model of the flow, the Partial Differential Equations (PDEs). The resulting expressions are then discretized. In the discrete approach, numerical approximations of the PDEs and functions, such as objective and constraints, are performed prior to the derivation of the adjoint problem. The formulation of an optimization problem in §2, illustrates some differences of implementation between the two methods. The two methods

are compared in [28], when applied to the optimization of a shape embedded in a viscous flow. Concerning the discrete approach, the reader can find in [31] the derivation of adjoint equations for elliptic problems. In several reports [21], [19] and [20], Giles presents the implementation, but no detailed derivations, of the discrete adjoint equations in industrial flow solvers.

The accuracy of gradients is considered a crucial issue in this work and this is the main reason to adopt the discrete approach. As noted above this choice binds the adjoint code to the flow solver together in the way that the adjoint equation is a transpose of the linearized discrete flow equation. This aspect is discussed in depth in this report. The flow solver is the code EDGE [12], which uses a discretization (§3) that recently has become popular for industrial problems with complex geometry.

Results of aerodynamic shape optimization are obtained by the method of discrete adjoint in [1, 14, 20, 26, 27, 28]. Only little is reported for the type of discretization used here. In contrast to the above references, the presentation below contains detailed derivations of the discrete adjoint equations and gradient expressions. For results on similar applications, but based on the continuous approach, the reader may refer to [3, 11, 15, 24, 34, 35].

Implementation issues may explain the recent interest for the automated production of derivatives using softwares of Automatic Differentiation (AD) such as Tapenade [16] (a package previously known as Odyssée), ADIFOR [5], TAMC [18], ADOL-C [22], and FAD [32]. In another reference [26], Odyssée was applied for viscous optimization based on Reynolds Averaged Navier–Stokes (RANS) equations for 2D applications, and on the Euler equations for the reduction of the wave drag in 3D. Limitations of the AD are due to huge memory requirements in the so-called reverse mode, which corresponds to an adjoint solution. However, the method may be used 'locally'. For example, Giles [20] obtained adjoints of the artificial dissipation fluxes by AD.

In addition to a gradient-based algorithm (see chapter 6) and adjoint equations for the calculation of gradients, our approach of shape optimization (see chapter 2) requires to use a mesh movement algorithm (§2.1), and a parameterization.

Parameterization is a universal feature of shape optimization. Many authors use piecewise polynomial interpolations of the shape, controlled by significantly fewer parameters than the number of nodes on the design boundary [1, 3, 6, 7, 8, 35]. Alternative methods are explored in [15], where the design boundary is decomposed in a basis of shape functions, in [34], where approximated second derivatives at each nodes are the control parameters, and in [27], where the displacements of the nodes on the shape are solution of a discrete smoothing equation.

Our approach is an extension of the one used in [27], and is motivated by the desire to have a detailed description of the design. The nodal displacements on the design boundary could be the control parameters but, if applied without restrictions, the smoothness of the shape would be destroyed in a few updates of the optimization algorithm. A parameterization strategy is devised in §5.2-5.3, in order to smooth out wiggles as well as to handle geometric constraints.

Results of drag minimization problems, obtained in 2D for the airfoil RAE 2822 [10], are reported in §6.1. In these applications, the objective is to reduce the wave drag C_D under the constraints that the lift C_L and the pitch moments C_M are maintained close to their initial design values and under constraints on the geometry such as leading-edge radius and volume.

A second application, reported in §6.3, is an ongoing project where we consider the Euler equations coupled with the boundary layer and parabolized stability equations. The aim is to delay transition, in order to reduce the viscous drag.

The method that is used to obtain a delay of the transition by shape optimization (§6.3) is the result of investigations presented in Paper A.

Chapter 2

The optimization problem

The numerical solution to a shape optimization problem requires approximations of the state equation (see §3.3). Therefore, the first section §2.1 presents the formulation of a typical problem of aerodynamic shape optimization in continuous form (P) and after discretization of the state equation (P_h). The subscript h is used to denote a grid parameter such as the largest diameter of any element in the mesh.

Thereafter, in §2.2, it is shown how the method of adjoint is applied, within the framework of the discretized problem (P_h), in order to calculate gradients.

2.1 Shape optimization

A typical problem of aerodynamic shape optimization minimizes an objective function J , for given constraints C_j , with respect to a parameterization \mathbf{y} of the shape Γ . Further restrictions may be imposed by choosing a set of *feasible designs* \mathcal{F} . The functions J and C_j are often surface integrals on the shape Γ that depend on the flow state denoted \mathbf{w} . Therefore we use the notations $J(\mathbf{w}, \Gamma)$ and $C_j(\mathbf{w}, \Gamma)$, for $1 \leq j \leq m_a$, to indicate that the objective J and the m_a constraints C_j depend explicitly on the flow state \mathbf{w} and on the shape Γ . For instance, in drag minimization problems, it is common that the objective is the drag coefficient, and that constraints are imposed on the lift and moment coefficients. The parameterization of the shape Γ in terms of a set of design variables \mathbf{a} is expressed

$$\mathcal{S}(\Gamma, \mathbf{a}) = \mathbf{0}, \quad (2.1)$$

which can be analytic expressions or a system of equations, for example. It is required that \mathcal{S} is continuously differentiable with respect to \mathbf{a} and Γ as

we use gradient-based methods for the optimization. The notation $C_j(\Gamma)$, for $m_a + 1 \leq j \leq m_a + m_g$, indicates that C_j depends on the shape only, which is the usual way to impose geometric constraints. The optimization problem may thus be formulated as

$$(P) \begin{cases} \hat{\mathbf{a}} = \arg \min_{\mathbf{a} \in \mathcal{F}} J(\mathbf{w}, \Gamma) & \text{subject to} \\ \mathcal{S}(\Gamma, \mathbf{a}) = \mathbf{0} , \\ \mathcal{A}(\mathbf{w}, \Gamma) = \mathbf{0} , & \text{on } \Omega_\Gamma \\ C_j(\mathbf{w}, \Gamma) \geq 0 , & 1 \leq j \leq m_a , \\ C_j(\Gamma) \geq 0 , & m_a + 1 \leq j \leq m_a + m_g , \end{cases}$$

where

$$\mathcal{A}(\mathbf{w}, \Gamma) = \mathbf{0} , \quad (2.2)$$

is the state equation, which defines a unique flow state \mathbf{w} for a given shape Γ , of the domain Ω_Γ .

In a numerical approximation of the optimization problem (P), see chapter 3, the domain Ω_Γ is replaced by a discretized bounded domain Ω_h . From the mesh we need the set of nodal coordinates \mathbf{X}_h . Note that the connectivities between the nodes are unchanged during a deformation of the mesh. The meshing process involves discretization of the shape Γ , denoted Γ_h .

In EDGE [12], the discretization of the state equations (Euler or RANS) makes use of the set of surface vectors \mathbf{n}_h , which are expressed from the nodal coordinates \mathbf{X}_h , and the connectivities, that is $\mathbf{n}_h \equiv \mathbf{n}_h(\mathbf{X}_h)$ (appendix A). The discrete state equation is

$$\mathcal{A}_h(\mathbf{w}_h, \mathbf{n}_h) = \mathbf{0} , \quad (2.3)$$

where \mathbf{w}_h is the discrete flow state, that is $\mathbf{w}_h \equiv \mathbf{w}_h(\mathbf{n}_h)$.

After discretization, the value of the functions J_h and C_{hj} are expressions that depend on the discrete state \mathbf{w}_h and on the discretized surface of the shape, a subset of \mathbf{n}_h . Therefore we use the notations $J_h(\mathbf{w}_h, \mathbf{n}_h)$ and $C_{hj}(\mathbf{w}_h, \mathbf{n}_h)$.

Similarly, \mathbf{a}_h denotes the parameterization of the discretized shape, a discrete analogue of \mathbf{a} . The parameterization \mathbf{a}_h controls the displacements \mathbf{y}_h of the nodes that are on the discretized shape Γ_h , via the mapping \mathcal{S}_h , a discrete analogue of \mathcal{S} ,

$$\mathcal{S}_h(\mathbf{y}_h, \mathbf{a}_h) = \mathbf{0} , \quad (2.4)$$

To retain mesh quality, the nodal coordinates \mathbf{X}_h must be altered when the displacements \mathbf{y}_h are changed. This change can be accomplished either by complete re-meshing or by a smooth propagation of the deformation from

the boundary into the interior of the mesh. We have chosen the latter approach since we use gradient-based optimization and thus require a smooth dependence, on the flow state, of the design variables. The propagation of deformation of the shape is accomplished by a smooth mesh movement mapping $\mathbf{y}_h \rightarrow \mathbf{X}_h$ and denoted

$$\mathcal{M}_h(\mathbf{X}_h, \mathbf{y}_h) = \mathbf{0}. \quad (2.5)$$

Thus, the formulation of problem (P), after discretization is:

$$(P_h) \left\{ \begin{array}{ll} \hat{\mathbf{a}}_h & = \arg \min_{\mathbf{a}_h \in \mathcal{F}_h} J_h(\mathbf{w}_h, \mathbf{n}_h(\mathbf{X}_h)) \quad \text{subject to} \\ \mathcal{S}_h(\mathbf{y}_h, \mathbf{a}_h) & = \mathbf{0}, \\ \mathcal{M}_h(\mathbf{X}_h, \mathbf{y}_h) & = \mathbf{0}, \\ \mathcal{A}_h(\mathbf{w}_h, \mathbf{n}_h(\mathbf{X}_h)) & = \mathbf{0}, \\ C_{hj}(\mathbf{w}_h, \mathbf{n}_h(\mathbf{X}_h)) & \geq \mathbf{0}, 1 \leq j \leq m_a, \\ C_{hj}(\mathbf{n}_h(\mathbf{X}_h)) & \geq \mathbf{0}, m_a + 1 \leq j \leq m_a + m_g, \end{array} \right.$$

where \mathcal{F}_h is a set of admissible parameterizations of the discretized shapes, a discrete analogue of \mathcal{F} .

2.2 Discrete adjoint in shape optimization

The implementation of adjoint equations in the CFD code EDGE is a crucial component of this project as it provides an efficient tool for computing gradients of the functions J_h and C_{hj} .

Let $f(\mathbf{w}_h, \mathbf{n}_h)$ denote any of the functions $J_h(\mathbf{w}_h, \mathbf{n}_h)$ and $C_{hj}(\mathbf{w}_h, \mathbf{n}_h)$. We denote by f_a the function

$$\mathbf{a}_h \rightarrow f(\mathbf{w}_h(\mathbf{n}_h(\mathbf{X}_h(\mathbf{y}_h(\mathbf{a}_h))))), \mathbf{n}_h(\mathbf{X}_h(\mathbf{y}_h(\mathbf{a}_h)))) ,$$

where $\mathbf{n}_h \equiv \mathbf{n}_h(\mathbf{X}_h)$ is obtained from \mathbf{a}_h by solving successively (2.4) and (2.5), and \mathbf{w}_h is obtained from \mathbf{n}_h by solving (2.3). It will be convenient to denote by f_n the function

$$\mathbf{n}_h \rightarrow f(\mathbf{w}_h(\mathbf{n}_h), \mathbf{n}_h) ,$$

where \mathbf{w}_h is obtained from \mathbf{n}_h by solving (2.3).

Suppose that \mathcal{F}_h is included in \mathcal{U} , a vector space equipped with the inner product $\langle \cdot, \cdot \rangle_U$. Assume that any of the functions J_h , C_{hj} , the discretized state equation (2.3), the deformation of the mesh (2.5), and the parameterization (2.4) are continuously differentiable. Thus, the gradient of f_a , denoted ∇f_a , is defined as follows

$$\delta f_a = \langle \nabla f_a, \delta \mathbf{a}_h \rangle_U . \quad (2.6)$$

for all $\delta \mathbf{a}_h$ in \mathcal{U} , where δf_a is the first variation of f_a , which is expressed as

$$\delta f_a = \lim_{\lambda \rightarrow 0} \frac{1}{\lambda} [f_a(\mathbf{a}_h + \lambda \delta \mathbf{a}_h) - f_a(\mathbf{a}_h)] . \quad (2.7)$$

The gradient ∇f_a is also known as the *reduced gradient* of f . Following chapters will show that ∇f_a is given by the following relations

$$\nabla f_a = - \left(\frac{\partial \mathcal{S}_h}{\partial \mathbf{a}_h} \right)^* \mathbf{y}_h^* , \quad (2.8)$$

$$\left(\frac{\partial \mathcal{S}_h}{\partial \mathbf{y}_h} \right)^* \mathbf{y}_h^* = - \left(\frac{\partial \mathcal{M}_h}{\partial \mathbf{y}_h} \right)^* \mathbf{X}_h^* , \quad (2.9)$$

$$\left(\frac{\partial \mathcal{M}_h}{\partial \mathbf{X}_h} \right)^* \mathbf{X}_h^* = - \left(\frac{d\mathbf{n}_h}{d\mathbf{X}_h} \right)^T \left[- \left(\frac{\partial \mathcal{A}_h}{\partial \mathbf{n}_h} \right)^* \mathbf{w}_h^* + \frac{\partial f}{\partial \mathbf{n}_h} \right] , \quad (2.10)$$

$$\left(\frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \right)^* \mathbf{w}_h^* = \frac{\partial f}{\partial \mathbf{w}_h} , \quad (2.11)$$

where $(\partial \mathcal{S}_h / \partial \mathbf{a}_h)^*$ and $(\partial \mathcal{S}_h / \partial \mathbf{y}_h)^*$ denote adjoints of the linearized operator \mathcal{S}_h with respect to \mathbf{a}_h and \mathbf{y}_h , respectively, $(\partial \mathcal{M}_h / \partial \mathbf{y}_h)^*$ and $(\partial \mathcal{M}_h / \partial \mathbf{X}_h)^*$, denote adjoints of the linearized operator \mathcal{M}_h with respect to \mathbf{y}_h and \mathbf{X}_h , respectively. Moreover $(\partial \mathcal{A}_h / \partial \mathbf{n}_h)^*$ and $(\partial \mathcal{A}_h / \partial \mathbf{w}_h)^*$, denote adjoints of the linearized operator \mathcal{A}_h with respect to \mathbf{n}_h and \mathbf{w}_h , respectively. The notation \mathbf{w}_h^* is used for the solution of the adjoint state equation (2.11), which is presented in detail in §4, \mathbf{X}_h^* is the solution of the adjoint mesh movement equation (2.10), for which details are given in §5.1, and \mathbf{y}_h^* is the solution of the adjoint parameterization equation, described in §5.2-5.3. In order to show that relation (2.8) holds, one must prove that

$$\nabla f_n = - \left(\frac{\partial \mathcal{A}_h}{\partial \mathbf{n}_h} \right)^* \mathbf{w}_h^* + \frac{\partial f}{\partial \mathbf{n}_h} , \quad (2.12)$$

that is, the gradient of the function f_n . This demonstration is given in appendix B, (B.66).

Chapter 3

State equation and discretization

This section presents the Euler equations §3.1, the finite-volume discretization §3.2 and, the edge-based calculation of the residuals as performed in EDGE §3.3. The discretization is based on a median dual grid which is briefly described in appendix A. For more details about this type of discretization we refer to [2].

3.1 Euler equations

The Euler equations express the conservation of mass, the conservation of momentum, and the conservation of energy, written here in integral form for an arbitrary fixed region V with boundary ∂V ,

$$\frac{\partial}{\partial t} \int_V \rho \, dX + \int_{\partial V} \rho \mathbf{u} \cdot \hat{\mathbf{n}} \, dS = 0, \quad (3.1)$$

$$\frac{\partial}{\partial t} \int_V \rho \mathbf{u} \, dX + \int_{\partial V} \rho \mathbf{u} \mathbf{u} \cdot \hat{\mathbf{n}} \, dS + \int_{\partial V} p \hat{\mathbf{n}} \, dS = \mathbf{0}, \quad (3.2)$$

$$\frac{\partial}{\partial t} \int_V E \, dX + \int_{\partial V} \mathbf{u} E \cdot \hat{\mathbf{n}} \, dS + \int_{\partial V} p \mathbf{u} \cdot \hat{\mathbf{n}} \, dS = 0, \quad (3.3)$$

where $\hat{\mathbf{n}}$ is the unit outward-oriented normal on ∂V , E is the total energy per unit volume, which for an ideal fluid is related to the temperature and the velocity by

$$E = \rho C_v T + \frac{1}{2} \rho \mathbf{u}^2. \quad (3.4)$$

A constitutive law for the gas, for instance the law of perfect gas

$$\frac{p}{\rho} = \mathcal{R}T, \quad (3.5)$$

yields

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\mathbf{u}^2. \quad (3.6)$$

Let \mathbf{f} be the 3-by-1 matrix of tensors

$$\mathbf{f} = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u} \otimes \mathbf{u} + Ip \\ \mathbf{u}(E + p) \end{pmatrix}, \quad (3.7)$$

and let the flow state \mathbf{w} be the 3-by-1 matrix of conservative variables

$$\mathbf{w} = \begin{pmatrix} \rho \\ \mathbf{m} \\ E \end{pmatrix}, \quad (3.8)$$

where

$$\mathbf{m} = \rho\mathbf{u}. \quad (3.9)$$

Let us note that the flow state in primitive variables is the 3-by-1 matrix of tensors

$$\mathbf{v} = \begin{pmatrix} \rho \\ \mathbf{u} \\ p \end{pmatrix}. \quad (3.10)$$

The dot product of \mathbf{f} (3.7) with any first-order tensor \mathbf{n} (not just normal vectors) may be defined by element-wise operations,

$$\mathbf{n} \cdot \mathbf{f} = \mathbf{f} \cdot \mathbf{n} = \begin{pmatrix} \rho\mathbf{u} \cdot \mathbf{n} \\ \rho\mathbf{u}\mathbf{u} \cdot \mathbf{n} + np \\ E\mathbf{u} \cdot \mathbf{n} + p\mathbf{u} \cdot \mathbf{n} \end{pmatrix}, \quad (3.11)$$

Using relations (3.4)-(3.11), the Euler equations (3.1)-(3.3) take the form

$$\frac{\partial}{\partial t} \int_V \mathbf{w} dX + \int_{\partial V} \mathbf{f} \cdot \hat{\mathbf{n}} dS = \mathbf{0}, \quad (3.12)$$

which, in steady state, simplifies to

$$\int_{\partial V} \mathbf{f} \cdot \hat{\mathbf{n}} dS = \mathbf{0}. \quad (3.13)$$

3.2 Finite-volume approximation

The finite-volume method implemented in EDGE is node-centered, that is, the state variables are computed at the nodes and the fluxes (3.7) are computed at the faces of control volumes 'centered' on the nodes, see figures (A.1)-(A.4).

Given a triangulation \mathcal{T}_h of the bounded domain of computation Ω_h , a dual grid is defined based on the elements of \mathcal{T}_h (see appendix A). The surfaces of the control volumes V_i are defined by the surface vectors \mathbf{n}_{ij} and \mathbf{n}_i , where \mathbf{n}_{ij} is associated to the edge \vec{ij} and, \mathbf{n}_i is associated to a node i on the boundary.

Equation (3.13) holds for all control volumes V in the domain of computation Ω_h , in particular for $V = V_i$, where V_i is the control volume at node i defined by the dual grid. Therefore applying the conservation of the fluxes (3.13) at each control volume yields a system of equations, one equation of conservation at each node,

$$\begin{aligned} \sum_{j \in \mathcal{N}_i} \mathbf{n}_{ij} \cdot \frac{1}{|\mathbf{n}_{ij}|} \int_{\partial V_{ij}} \mathbf{f} \, dS &= 0 \quad \forall i \in \mathcal{V}(\Omega_h), \\ \sum_{j \in \mathcal{N}_i} \mathbf{n}_{ij} \cdot \frac{1}{|\mathbf{n}_{ij}|} \int_{\partial V_{ij}} \mathbf{f} \, dS + \mathbf{n}_i \cdot \frac{1}{|\mathbf{n}_i|} \int_{\partial V_{i0}} \mathbf{f} \, dS &= 0 \quad \forall i \in \mathcal{V}(\partial\Omega_h), \end{aligned} \quad (3.14)$$

where ∂V_{ij} denotes the surface, of the control volume V_i , associated with the edge \vec{ij} , and ∂V_{i0} is the surface, of the volume V_i , that is on the boundary $\partial\Omega_h$. The set of nodes that are connected to node i with an edge \vec{ij} is denoted \mathcal{N}_i , $\mathcal{V}(\Omega_h)$ denotes the set of nodes in the interior of Ω_h , and $\mathcal{V}(\partial\Omega_h)$ is the set of nodes on the boundary $\partial\Omega_h$.

The integrand in (3.14) is approximated by constant fluxes at the control surfaces, denoted \mathbf{f}_{ij} for a flux on a control surface in the interior of Ω_h and \mathbf{f}_i^{bc} for a flux on a control surface on the boundary $\partial\Omega_h$, which yields

$$\begin{aligned} \sum_{j \in \mathcal{N}_i} [\mathbf{n}_{ij} \cdot \mathbf{f}_{ij} + \mathbf{d}_{ij}] &= 0 \quad \forall i \in \mathcal{V}(\Omega_h), \\ \sum_{j \in \mathcal{N}_i} [\mathbf{n}_{ij} \cdot \mathbf{f}_{ij} + \mathbf{d}_{ij}] + \mathbf{n}_i \cdot \mathbf{f}_i^{\text{bc}} &= 0 \quad \forall i \in \mathcal{V}(\partial\Omega_h), \end{aligned} \quad (3.15)$$

In EDGE code the fluxes, in the interior of the domain of computation, are calculated as

$$\mathbf{f}_{ij} = \frac{1}{2} (\mathbf{f}_i + \mathbf{f}_j) \quad \text{with} \quad \mathbf{f}_i = \mathbf{f}(\mathbf{w}_i), \quad (3.16)$$

with \mathbf{w}_i denoting the flow state (3.8) at node i . The notation \mathbf{d}_{ij} is used for artificial dissipation fluxes that are detailed in the next section §3.3. At a wall $\mathbf{f}_i^{\text{bc}} = \mathbf{f}_i^w$, where

$$\mathbf{f}_i^w = \begin{pmatrix} \mathbf{0} \\ Ip_i \\ \mathbf{0} \end{pmatrix}. \quad (3.17)$$

At a farfield boundary $\mathbf{f}_i^{\text{bc}} = \mathbf{f}_i^\infty$, where \mathbf{f}_i^∞ is calculated by computing \mathbf{f} (3.7) using, either the primitive farfield data \mathbf{v}_∞ , for incoming characteristics, or

an extrapolation for characteristics leaving the domain of computation:

$$\begin{aligned} \mathbf{f}_i^\infty &= \mathbf{f}(\mathbf{v}_i^c(\hat{\mathbf{n}}_i)) , \\ \mathbf{v}_i^c(\hat{\mathbf{n}}_i) &= \mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{H}(\lambda_i) \mathbf{L}^{-1}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{v}_i \\ &\quad + \mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) (\mathbf{I} - \mathbf{H}(\lambda_i)) \mathbf{L}^{-1}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{v}_\infty , \end{aligned} \quad (3.18)$$

where $\mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty)$ is a matrix of left eigenvectors that diagonalizes the Jacobian matrix of the flux in primitive variables along the outward-directed unit normal $\hat{\mathbf{n}}_i$, $\mathbf{H}(\lambda_i)$ is a diagonal matrix whose diagonal is 0 for negative eigenvalues and 1 for positive ones, and \mathbf{I} is the identity matrix.

Note that \mathbf{f}_i^w (3.17) is obtained from \mathbf{f}_i (3.7) by assuming that the slip condition holds, that is, at a node i on the boundary $\partial\Omega_h$,

$$\mathbf{u}_i \cdot \hat{\mathbf{n}}_i = 0 . \quad (3.19)$$

This assumption is not equivalent to imposing $\mathbf{u}_i \cdot \hat{\mathbf{n}}_i = 0$. The way boundary conditions are imposed in the state equation influences the derivation of the adjoint equations. This aspect is further discussed by Giles [20].

3.3 Edge-based residual assembly

The program EDGE [12] solves equations (3.15) by explicit time integration of the system

$$V_i \frac{d\mathbf{w}_i}{dt} + \mathbf{R}_i = 0, \quad \forall i \in \mathcal{V}(\bar{\Omega}_h), \quad (3.20)$$

until the residuals \mathbf{R}_i , at each node, vanishes within some tolerance. Convergence is accelerated by local time stepping, multigrid, and implicit residual smoothing. The residuals \mathbf{R}_i are assembled by a single loop over all edges and all boundary nodes :

For each edge \vec{ij} :

$$\begin{aligned} \mathbf{R}_i &\leftarrow \mathbf{R}_i + \mathbf{n}_{ij} \cdot \mathbf{f}_{ij} + \mathbf{d}_{ij} \\ \mathbf{R}_j &\leftarrow \mathbf{R}_j - \mathbf{n}_{ij} \cdot \mathbf{f}_{ij} - \mathbf{d}_{ij} \end{aligned} \quad (3.21)$$

For each boundary node i :

$$\mathbf{R}_i \leftarrow \mathbf{R}_i + \mathbf{n}_i \cdot \mathbf{f}_i^{\text{bc}}$$

where \mathbf{d}_{ij} is an artificial dissipation flux, a blend of second- and fourth-order differences of Jameson type

$$\mathbf{d}_{ij} = \epsilon_2^{ij} (\mathbf{w}_i - \mathbf{w}_j) + (\epsilon_4^i \nabla^2 \mathbf{w}_i - \epsilon_4^j \nabla^2 \mathbf{w}_j), \quad (3.22)$$

with

$$\nabla^2 \mathbf{w}_i = \sum_{k \in \mathcal{N}_i} (\mathbf{w}_k - \mathbf{w}_i) \quad \forall i \in \mathcal{V}(\bar{\Omega}_h). \quad (3.23)$$

The second-order dissipation is active where pressure gradients are large to prevent oscillations in the vicinity of shocks. The fourth-order dissipation is meant to remove oscillating solutions from grid point to grid point while preserving the second-order accuracy of the central scheme away from the shocks.

We give here details of the flux of the 2nd order artificial dissipation associated with edge $i\vec{j}$ because it is used later in the derivation of the adjoint equations:

$$\mathbf{d}_{ij}^{2\text{nd}} = \epsilon_2^{ij} (\mathbf{w}_i - \mathbf{w}_j), \quad (3.24)$$

with

$$\epsilon_2^{ij} = \epsilon_2 \max(\text{psw}_i, \text{psw}_j) N_{ij} F_{ij} \quad (3.25)$$

where ϵ_2 is a user defined parameter, psw_i is the pressure sensor at node i

$$\text{psw}_i = \frac{|\sum_{j \in \mathcal{N}_i} (p_i - p_j)|}{\sum_{j \in \mathcal{N}_i} (p_i + p_j) + \epsilon}, \quad \epsilon \ll \epsilon_{\text{machine}}, \quad (3.26)$$

N_{ij} is a grid parameter which depends on the number of edges at node i and j , and

$$F_{ij} = 4 \frac{\beta_{ij} \beta_{ji}}{\beta_{ij} + \beta_{ji}} R_{ij}, \quad (3.27)$$

with

$$\begin{aligned} \beta_{ij} &= \left(\frac{\lambda_i}{4R_{ij}} \right)^{1/2}, \\ R_{ij} &= \frac{1}{2} (r_{ij} + r_{ji}), \\ r_{ij} &= |\mathbf{u}_i \cdot \mathbf{n}_{ij}| + c_i |\mathbf{n}_{ij}|, \end{aligned} \quad (3.28)$$

where c_i is the sound speed at node i , and λ_i is an integrated spectral radius given by the following relations

$$\begin{aligned} \lambda_i &= \frac{1}{2} \sum_{j \in \mathcal{N}_i} [|\mathbf{u}_i + \mathbf{u}_j| \cdot \mathbf{n}_{ij}| + (c_i + c_j) |\mathbf{n}_{ij}|] \quad \forall i \in \mathcal{V}(\Omega_h), \\ \lambda_i &= \frac{1}{2} \sum_{j \in \mathcal{N}_i} [|\mathbf{u}_i + \mathbf{u}_j| \cdot \mathbf{n}_{ij}| + (c_i + c_j) |\mathbf{n}_{ij}|] \\ &\quad + |\mathbf{u}_i \cdot \mathbf{n}_i| + c_i |\mathbf{n}_i| \quad \forall i \in \mathcal{V}(\partial\Omega_h). \end{aligned} \quad (3.29)$$

Let us note some properties from definitions (3.25)-(3.29). Despite the non-symmetry of some terms $r_{ij} \neq r_{ji}$, $\beta_{ij} \neq \beta_{ji}$, we have symmetry for the artificial viscosity coefficient ϵ_2^{ij} since

$$R_{ij} = R_{ji}, \quad \beta_{ij} \beta_{ji} = \beta_{ji} \beta_{ij}, \quad \beta_{ij} + \beta_{ji} = \beta_{ji} + \beta_{ij}, \quad (3.30)$$

yields symmetry for F_{ij} (3.27), that is $F_{ij} = F_{ji}$, and $N_{ij} = N_{ji}$ as well as $\max(\text{psw}_i, \text{psw}_j) = \max(\text{psw}_j, \text{psw}_i)$ finally gives

$$\epsilon_2^{ij} = \epsilon_2^{ji}. \quad (3.31)$$

Chapter 4

Adjoint equation and reduced gradient

For a given discrete approximation of a flow model, such as the Euler equations discretized in the code EDGE [12], an adjoint system of equations is defined for each objective function or state constraint $f(\mathbf{w}_h, \mathbf{n}_h)$ for which the shape gradient is needed. Section §4.1 introduces the method of adjoints based on discrete sensitivities. The adjoint equations that are derived in appendix B, based on the discrete sensitivities of the Euler equations, are presented in §4.2. In section §4.3, the expression of the reduced gradient ∇f_n is given, as defined in §2.2.

4.1 Discrete sensitivities

To introduce the general idea behind the adjoint approach applied to the discretized optimization problem (P_h) presented in chapter 2, we consider first a simpler problem in the spirit of Giles & Pierce [21]. Let f be a linear function of \mathbf{z} ,

$$f(\mathbf{z}) = \mathbf{g}^T \mathbf{z}, \quad (4.1)$$

with $\mathbf{g} \in \mathbb{R}^n$ given and $\mathbf{z} \in \mathbb{R}^n$ subject to the state equation

$$\mathbf{A}\mathbf{z} = \mathbf{N}\mathbf{a}, \quad (4.2)$$

with $\mathbf{a} \in \mathbb{R}^m$, $\mathbf{A} \in \mathbb{R}^{n \times n}$, and $\mathbf{N} \in \mathbb{R}^{n \times m}$.

Assume that \mathbf{A} is nonsingular. The reduced gradient of f , that is, the gradient of the mapping $\mathbf{a} \rightarrow f(\mathbf{z}(\mathbf{a}))$, denoted ∇f_a , may be obtained by solving the *sensitivity equations* of the state: given a variation of the control variable $\delta \mathbf{a}_k$, corresponding variation of the state $\delta \mathbf{z}_k$ is defined as the solution

to the sensitivity equations

$$\mathbf{A} \delta \mathbf{z}_k = \mathbf{N} \delta \mathbf{a}_k, \quad (4.3)$$

which enables us to express the variation of the function f

$$\delta f = \mathbf{g}^T \delta \mathbf{z} \equiv \mathbf{g}^T (\mathbf{A}^{-1} \mathbf{N} \delta \mathbf{a}_k). \quad (4.4)$$

Therefore, solving the sensitivity equations, once for each component k of the control \mathbf{a} , yields the gradient ∇f_a . However, rewriting (4.4) as

$$\delta f = (\mathbf{N}^T \mathbf{A}^{-T} \mathbf{g})^T \delta \mathbf{a}_k, \quad (4.5)$$

reveals that replacing $\mathbf{A}^{-T} \mathbf{g}$ in (4.5) by the adjoint state \mathbf{z}^* , defined as the solution to

$$\mathbf{A}^T \mathbf{z}^* = \mathbf{g}, \quad (4.6)$$

gives an expression for ∇f_a ,

$$\nabla f_a = \mathbf{N}^T \mathbf{z}^*. \quad (4.7)$$

The cost for computing the gradient by expression (4.7) is one costate solution (4.6), instead of m solutions of the sensitivity equations (4.3) when expression (4.4) is used.

The generalization to the nonlinear state equation (2.3) and nonlinear functions f , such as the penalization functions used in chapter 6, is straightforward:

- \mathbf{g} is the linearization of the function f with respect to the state variable, and will be denoted $\partial f / \partial \mathbf{w}_h$,
- \mathbf{A} is the Jacobian matrix of the system of state equations with respect to the state variable, and will be denoted $\partial \mathcal{A}_h / \partial \mathbf{w}_h$,
- \mathbf{N} expresses the sensitivity of the system of state equations with respect to the control variable, which, in the formulation of EDGE, see chapter 3, is the set of surface vectors \mathbf{n}_h . It will be denoted $\partial \mathcal{A}_h / \partial \mathbf{n}_h$.

However, many intermediate expressions need to be differentiated when linearizing \mathcal{A}_h . These expressions together with additional notations are defined in §B.1. The sensitivity equations are derived in §B.2. Note that the linearized operators are not expressed as matrices, therefore, identification of the adjoint operator, the analogue to \mathbf{A}^T above, requires the definition of an inner product. The transposition operation is defined in §B.1 and is used in §B.2.1 to identify the adjoint operator. For functions f of the variables \mathbf{w}_h and \mathbf{n}_h , denoted $f(\mathbf{w}_h, \mathbf{n}_h)$, subject to the state equation (2.3), the reduced gradient ∇f_n is calculated from the solution of adjoint equations defined in §B.2.2.

4.2 Edge-based residual assembly

The adjoint equations (B.61) can be solved by explicit time integration of the system

$$V_i \frac{d\mathbf{w}_i^*}{dt} + \mathbf{R}_i^* = \mathbf{0}, \quad \forall i \in \mathcal{V}(\bar{\Omega}_h) \quad (4.8)$$

until the residuals \mathbf{R}_i^* vanish within some tolerance. As when solving the system of discretized Euler equations (3.20), convergence may be accelerated by local time stepping, multigrid, and implicit residual smoothing. It is a result of the derivation carried out in appendix B that the adjoint residual, \mathbf{R}^* , may be assembled by a single loop over all edges, all boundary nodes, and all nodes where f depends on \mathbf{w}_i , denoted $\text{supp } f$:

For each edge \vec{ij} :

$$\begin{aligned} \mathbf{R}_i^* &\leftarrow \mathbf{R}_i^* + \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T \frac{(\mathbf{w}_i^* - \mathbf{w}_j^*)}{2} + \mathbf{d}_{ij}^* \\ \mathbf{R}_j^* &\leftarrow \mathbf{R}_j^* + \left[\frac{\partial(\mathbf{f}_j \cdot \mathbf{n}_{ji})}{\partial \mathbf{w}_j} \right]^T \frac{(\mathbf{w}_j^* - \mathbf{w}_i^*)}{2} + \mathbf{d}_{ji}^* \end{aligned} \quad (4.9)$$

For each boundary node i :

$$\mathbf{R}_i^* \leftarrow \mathbf{R}_i^* + \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right]^T \mathbf{w}_i^*$$

For each node i in $\text{supp } f$:

$$\mathbf{R}_i^* \leftarrow \mathbf{R}_i^* - \left(\frac{\partial f}{\partial \mathbf{w}_i} \right)^T$$

The Jacobian matrix $\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij}) / \partial \mathbf{w}_i$ is given in appendix B. The adjoint \mathbf{d}^* of the artificial dissipation fluxes \mathbf{d} is here obtained by freezing the artificial viscosities, that is, the differentiation of ϵ_2^{ij} and ϵ_4^i with respect to \mathbf{w}_h is supposed to give rise to terms that can be neglected. The fully discrete adjoint of the artificial dissipation fluxes is given in appendix B.

The Jacobian of the fluxes applied at farfield boundaries assume that there are no vanishing eigenvalues, (see appendix B):

$$\begin{aligned} \frac{\partial(\mathbf{f}_i^\infty \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} &= \mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{H}(\lambda_i) \mathbf{L}^{-1}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{M}^{-1}(\mathbf{v}_i), \\ \mathbf{M}^{-1}(\mathbf{v}_i) &= \left(\frac{d\mathbf{v}_i}{d\mathbf{w}_i} \right). \end{aligned} \quad (4.10)$$

The Jacobian of the Euler wall flux function is expressed as

$$\frac{\partial(\mathbf{f}_i^w \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} = (\gamma - 1) \begin{pmatrix} \frac{1}{2} |\mathbf{u}_i|^2 \\ -\mathbf{u}_i \\ 1 \end{pmatrix}. \quad (4.11)$$

4.3 Reduced gradient

The solution \mathbf{w}_h^* of the adjoint problem described above is shown, in appendix B, to give rise to the following expression, for an arbitrary variation of the surface vectors $\delta \mathbf{n}_h$:

$$\delta f_n = \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n - \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega})} (\mathbf{w}_i^* - \mathbf{w}_j^*)^T \mathbf{f}_{ij} \cdot \delta \mathbf{n}_{ij} - \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{w}_i^{*T} \mathbf{f}_i^{\text{bc}} \cdot \delta \mathbf{n}_i, \quad (4.12)$$

when the variation of the artificial viscosities, with respect to a variation of the surface vectors, is neglected.

Chapter 5

Mesh movement and shape parameterizations

5.1 Mesh displacements

In the discrete approach (chapter 2), computing the sensitivities with respect to the nodal displacements \mathbf{X}_h is part of the gradient calculation (2.9)-(2.10). A complete remeshing of the domain of computation consists of calculating a new vector of nodal coordinates \mathbf{X}_h from an updated design, defined by the displacements \mathbf{y}_h . It defines a mapping $\mathbf{y}_h \rightarrow \mathbf{X}_h$ that may not be continuously differentiable, and may not be known by the user (black box). Therefore, in shape optimization, the update of the mesh is commonly handled by a specialized algorithm. General schemes are needed when using general unstructured meshes. Such algorithms are based, for example, on a spring analogy or on elliptic smoothing. Both techniques are investigated in [4]. In our investigation, we used structured meshes, which allows grid displacements defined by a simple formula. Therefore, an affine mapping can be defined as

$$\mathbf{X}^k = \mathbf{X}^0 + \mathbf{L}\mathbf{y}^k, \quad (5.1)$$

where $\mathbf{L} \in \mathbb{R}^{2N \times n}$ is a constant coefficients matrix, $\mathbf{y}^k \in \mathbb{R}^n$ is the vector of displacements that define the design k , and $\mathbf{X}^k \in \mathbb{R}^{2N}$ defines the mesh at design k (N is the total number of nodes in the grid). All displacements are with respect to a reference mesh, defined by its vector of nodal coordinates \mathbf{X}^0 .

Given the gradient ∇f_X of a function $f_X(\mathbf{X}_h)$ that may be defined as in appendix A, we denote by f_y the function

$$\mathbf{y}_h \rightarrow f_X(\mathbf{X}_h(\mathbf{y}_h)) .$$

The gradient of f_y is given by

$$\nabla f_y = \mathbf{L}^T \nabla f_X, \quad (5.2)$$

which is a consequence of the chain rule.

5.2 Unconstrained parameterization

This section details the mapping \mathcal{S}_h (2.4). The vector of displacements $\mathbf{y}_h = \{y_i \text{ for } 1 \leq i \leq n\}$ is defined by the relation between points located on the reference shape and points located on the displaced shape:

$$\mathbf{x}_i = \mathbf{x}_i^0 + y_i \hat{\mathbf{n}}_i^0, \quad \text{for } i \in \mathcal{V}(\partial\Omega_s), \quad (5.3)$$

where \mathbf{x}_i and \mathbf{x}_i^0 are nodes on the parameterized shape and the reference shape ($\partial\Omega_s$), respectively, and $\hat{\mathbf{n}}_i^0$ is a unit outward normal of the reference shape at node \mathbf{x}_i^0 . A well established technique in geometric modeling is the use of cubic splines (for curves) or NURBS (for curves or surfaces). For this type of parameterization the degree of differentiability is guaranteed by the interpolant. A second feature of this technique is the possibility to reduce the number of parameters of design while allowing for interesting shapes.

However, the use of an adjoint approach for the calculation of the gradient of a function f decouples the cost of the gradient evaluation from the number of parameters of design. Furthermore, the gradient based algorithm may generate strong oscillations in the sequence of control parameters, independently from the parameterization. Therefore, piecewise polynomial interpolation techniques, such as B-splines, may not avoid wavy shapes as experienced in [17].

Our technique is similar to the one proposed in [27]. The vector of displacements \mathbf{y}_h is chosen as the solution to a Poisson problem

$$\mathbf{A}_s \mathbf{y}_h = \mathbf{M}_s \mathbf{a}_h, \quad (5.4)$$

where \mathbf{A}_s is the (nonsingular) stiffness matrix associated with the discretization of the Laplace operator, and \mathbf{M}_s is a mass matrix.

Following the example in §4.1, given a function $f_y(\mathbf{y})$, we denote f_a the function

$$\mathbf{a}_h \rightarrow f_y(\mathbf{y}_h(\mathbf{a}_h)).$$

The gradient of f_a is obtained by solving an adjoint of the Poisson problem

$$\mathbf{A}_s^T \mathbf{y}_h^* = \nabla f_y, \quad (5.5)$$

which yields

$$\nabla f_a = \mathbf{M}_s^T \mathbf{y}_h^*. \quad (5.6)$$

5.3 Constrained parameterization

In this section, a quadratic program (QP) is formulated in order to extend the effect of smoothing of the Laplace operator, presented above, to constrained designs (\mathbf{y}_h).

In discrete, as well as in continuous form, the solution to the Poisson problem can be obtained by minimizing a quadratic functional. That is, the solution of (5.4), can also be defined as

$$\mathbf{y}_h = \min_{\mathbf{v}_h \in \mathbb{R}^n} \frac{1}{2} \mathbf{v}_h^T \mathbf{A}_s \mathbf{v}_h - \mathbf{v}_h^T \mathbf{M}_s \mathbf{a}_h. \quad (5.7)$$

Adding m_g linear constraints on the displacements defined by (5.7) enables us to control geometrical properties. The addition of constraints to (5.7) yields the following quadratic program (QP)

$$\mathbf{y}_h = \begin{cases} \min_{\mathbf{v}_h \in \mathbb{R}^n} \frac{1}{2} \mathbf{v}_h^T \mathbf{A}_s \mathbf{v}_h - \mathbf{v}_h^T \mathbf{M}_s \mathbf{a}_h, \\ \mathbf{C}^T \mathbf{v}_h = \mathbf{b} \end{cases}, \quad (5.8)$$

where \mathbf{C} is a matrix whose columns are the gradients of the constraints imposed on the displacements (in $\mathbb{R}^{n \times m_g}$) and \mathbf{b} is the vector of values of the constraints (in \mathbb{R}^{m_g}).

It is a known result of optimization theory [29] that the solution \mathbf{y}_h of the QP (5.8) is obtained by solving the Karush-Kuhn-Tucker (KKT) system

$$\begin{pmatrix} \mathbf{A}_s & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y}_h \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{M}_s \mathbf{a}_h \\ -\mathbf{b} \end{pmatrix}, \quad (5.9)$$

where $\lambda \in \mathbb{R}^{m_g}$ is a vector of Lagrange multipliers. The system (5.9) is the first order optimality conditions for the problem (5.8).

Expression (5.9), together with our assumptions on \mathbf{A}_s and \mathbf{C} , defines a nonsingular linear mapping

$$\begin{aligned} \mathbb{R}^n \times \mathbb{R}^{m_g} &\rightarrow \mathbb{R}^n \times \mathbb{R}^{m_g} \\ \{\mathbf{a}_h, \mathbf{b}\} &\rightarrow \{\mathbf{y}_h(\mathbf{a}_h, \mathbf{b}), \lambda(\mathbf{a}_h, \mathbf{b})\}. \end{aligned} \quad (5.10)$$

As compared with the example in §4.1, it will be convenient to extend the definition of a given function $f_y(\mathbf{y})$ to a function $f_{y,\lambda}(\mathbf{y}, \lambda)$ in order to separate the Lagrange multipliers and associated geometric constraints from the design variables. We denote by $f_{a,b}$ the function

$$\{\mathbf{a}_h, \mathbf{b}\} \rightarrow f_{y,\lambda}(\mathbf{y}_h(\mathbf{a}_h, \mathbf{b}), \lambda(\mathbf{a}_h, \mathbf{b})).$$

Following the same derivation as in §4.1, the gradient of $f_{a,b}$ is obtained through the following relations:

$$\begin{pmatrix} \mathbf{A}_s^T & -\mathbf{C} \\ -\mathbf{C}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y}_h^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} \nabla f_y \\ \mathbf{0} \end{pmatrix}, \quad (5.11)$$

and

$$\nabla f_{a,b} = \begin{pmatrix} \mathbf{M}_s^T & 0 \\ 0 & -\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{y}_h^* \\ \lambda^* \end{pmatrix}, \quad (5.12)$$

which yields gradients for the functions f_a and f_b defined in the usual way:

$$\nabla f_a = \mathbf{M}_s^T \mathbf{y}_h^* \quad \text{and} \quad \nabla f_b = -\lambda^*. \quad (5.13)$$

Chapter 6

Applications

The applications presented here are shape optimization problems that are formulated in the aim to reduce the drag on an airfoil, starting at the design of the RAE 2822 [10]. The constraints are to maintain the lift and the pitch coefficients close to the values of the RAE 2822, as well as the general geometrical characteristics of the RAE 2822.

The first test case uses inviscid flow calculations and concerns wave-drag minimization, that is, reduction of the drag caused by gas dynamic effects that occur close to the speed of sound. Several tests are carried out using this approach in §6.1 and §6.2.

In a second test, a strategy has been devised in order to simultaneously reduce the drag due to gas dynamic effects and the drag due to the presence of an attached, turbulent boundary layer. Here, we use inviscid flow calculations, followed by an analysis of the growth of disturbances in the boundary layer that may trigger the laminar-to-turbulent transition of the flow in the boundary layer. One result, taken from Paper A, is given in §6.3.

All applications below have constraints, which are included in a modified objective function by adding penalty terms to the function that is minimized. The optimization algorithm is a limited memory quasi-Newton algorithm for unconstrained problems [9].

We summarize below common data for the mesh, Mach number M , angle of attack α and corresponding lift C_L^0 and pitch moment C_M^0 coefficients, calculated from solutions of the Euler equations.

- Coarse mesh: C-type, 3412 nodes with 112 nodes on the airfoil,
 - $M_1 = 0.734$, $\alpha_1 = 2.19$ degrees, $C_{L_1}^0 = 0.801$, $C_{M_1}^0 = 0.328$
 - $M_2 = 0.754$, $\alpha_2 = 1.68$ degrees, $C_{L_2}^0 = 0.737$, $C_{M_2}^0 = 0.329$

- $M_3 = 0.68$, $\alpha_3 = 1.35$ degrees, $C_{L3}^0 = 0.560$, $C_{M3}^0 = 0.245$
- Medium mesh: C-type, 13352 nodes with 224 nodes on the airfoil,
 - $M_1 = 0.734$, $\alpha_1 = 2.19$ degrees, $C_{L1}^0 = 0.838$, $C_{M1}^0 = 0.342$
 - $M_2 = 0.754$, $\alpha_2 = 1.68$ degrees, $C_{L2}^0 = 0.772$, $C_{M2}^0 = 0.345$
 - $M_3 = 0.68$, $\alpha_3 = 1.35$ degrees, $C_{L3}^0 = 0.582$, $C_{M3}^0 = 0.254$

6.1 Wave drag optimization

The aim is to reduce the wave drag at Mach number M_1 while keeping the initial lift and pitch moment coefficients. A suitable objective function is

$$J = \lambda_{11} C_{D1} + \frac{1}{2} \lambda_{21} (C_{L1} - C_{L1}^0)^2 + \frac{1}{2} \lambda_{31} (C_{M1} - C_{M1}^0)^2, \quad (6.1)$$

where C_{D1} , C_{L1} and C_{M1} are the drag, lift and pitch moment coefficients, at Mach number M_1 . Here,

$$\begin{aligned} C_{D1} &= \sum_{i \in \mathcal{V}(\partial\Omega_w)} \frac{p_i \mathbf{n}_i \cdot \mathbf{d}_{D1}}{\frac{1}{2} \rho_\infty v_\infty^2 S_{\text{ref}}}, \\ C_{L1} &= \sum_{i \in \mathcal{V}(\partial\Omega_w)} \frac{p_i \mathbf{n}_i \cdot \mathbf{d}_{L1}}{\frac{1}{2} \rho_\infty v_\infty^2 S_{\text{ref}}}, \\ C_{M1} &= \sum_{i \in \mathcal{V}(\partial\Omega_w)} \frac{p_i \mathbf{d}_{M1} \cdot (\mathbf{x}_i - \mathbf{O}_{\text{ref}}) \times \mathbf{n}_i}{\frac{1}{2} \rho_\infty v_\infty^2 S_{\text{ref}} L_{\text{ref}}}. \end{aligned} \quad (6.2)$$

where \mathbf{d}_{D1} is a unit vector in the direction of the farfield velocity, \mathbf{d}_{L1} is a unit vector orthogonal to \mathbf{d}_{D1} and, \mathbf{d}_{M1} is a unit vector orthogonal to \mathbf{d}_{D1} and \mathbf{d}_{L1} . The scaling factors $\{\lambda_{11}, \lambda_{21}, \lambda_{31}\}$ are calculated as

$$\lambda_{11} = \frac{1}{10 C_{D1}^0}, \quad \lambda_{21} = \frac{10}{(C_{L1}^0)^2}, \quad \lambda_{31} = \frac{10}{(C_{M1}^0)^2}, \quad (6.3)$$

which were determined in preliminary tests.

The function J (6.1) is minimized in three optimization tests:

- 'Coarse-Vol': coarse mesh with constant volume constraint.
- 'Medium-Vol': medium mesh with constant volume constraint.
- 'Medium-LE': medium mesh with constant volume constraint and a fixed part of the airfoil, around the leading edge, between $x/c = 0$ and $x/c = 0.043$.

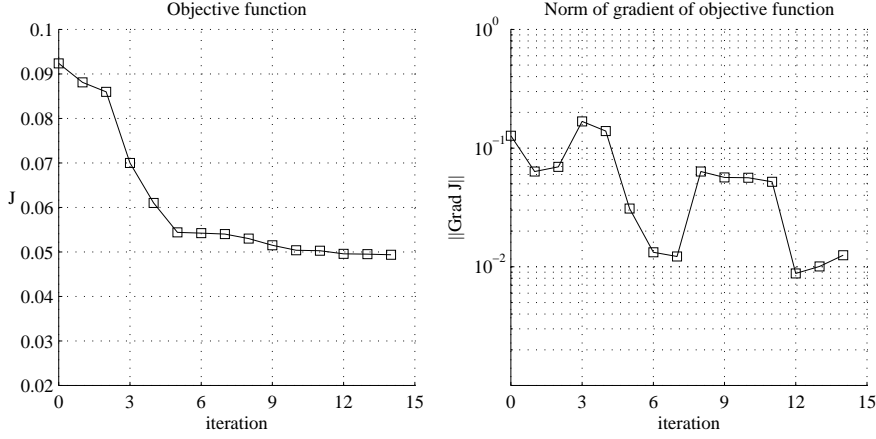


Figure 6.1: Case Medium-LE: objective function and gradient norm.

The results obtained at a single Mach number (M_1) are summarized in Table 6.1.

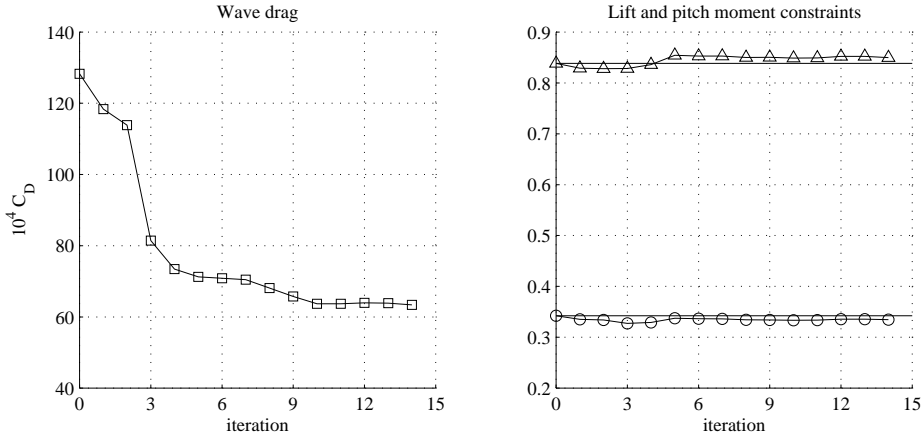


Figure 6.2: Case Medium-LE: wave drag (square-solid), lift (triangle-solid) and pitch moment (circle-solid) coefficients. The values at initial design are indicated by horizontal lines (solid).

The results obtained for the case 'Medium-LE' are shown on Figures 6.1-6.3. The history of the optimization is shown in Figure 6.1. Figure 6.2 shows that the wave drag is reduced by 67 drag-counts (10^{-4}) while the lift and pitch moment are within 1.5 % and 2 %, of their initial values. The reduction of the wave drag is achieved (Figure 6.3) by a smoothing of the shock wave. The changes on the geometry of the airfoil are a reduction of the thickness on the first half of the airfoil, between the leading edge and half the chord

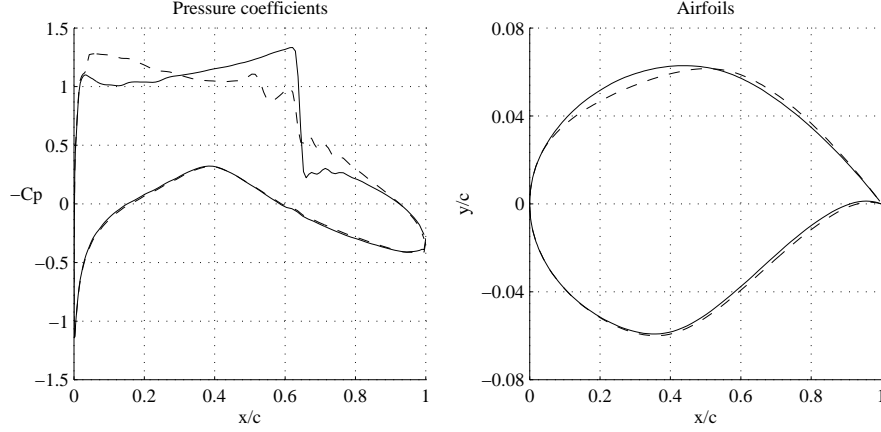


Figure 6.3: Case Medium-LE: pressure coefficients and shapes at initial (solid) and final design (dash).

length, and an increase of the thickness on the second half of the airfoil. The optimization process has not converged until it reached very small values for the norm of the gradient $\|\nabla J\|$ (see Figure 6.1) probably due to non-exact gradients (see Table 2 in Paper A). The designs, pressure distribution are similars for the two other tests. The small violation of the constraints is also a general trend of these tests.

	C_D	C_L	C_M
Coarse (Initial)	1.4×10^{-2}	8.01×10^{-1}	3.28×10^{-1}
Coarse-Vol	7.5×10^{-3}	7.95×10^{-1}	3.28×10^{-1}
Medium (Initial)	1.3×10^{-2}	8.38×10^{-1}	3.42×10^{-1}
Medium-Vol	5.6×10^{-3}	8.33×10^{-1}	3.42×10^{-1}
Medium-LE	6.3×10^{-3}	8.5×10^{-1}	3.35×10^{-1}

Table 6.1: Summary of results of single point optimization.

6.2 Multi-point wave drag optimization

The second type of tests concerns multi-point optimization. The aim is to minimize a weighted sum of the drags obtained at the three Mach numbers M_1 , M_2 and M_3 , that is $W_1 C_{D1} + W_2 C_{D2} + W_3 C_{D3}$. The weights are $W_1 = 2$, $W_2 = 1$, and $W_3 = 1$, giving the priority to the reduction of the drag at the "cruise" Mach number M_1 . It is also sought to maintain, at each design point i , the lift and pitch moment coefficients. In order to achieve this, an

objective function is designed as

$$J = \sum_{1 \leq i \leq 3} \left\{ W_i \lambda_{1i} C_{Di} + \frac{1}{2} \lambda_{2i} (C_{Li} - C_{Li}^0)^2 + \frac{1}{2} \lambda_{3i} (C_{Mi} - C_{Mi}^0)^2 \right\}, \quad (6.4)$$

where the coefficients $\{\lambda_{1i}, \lambda_{2i}, \lambda_{3i}\}$ are calculated in the same fashion as for the single point optimization (6.3). The function J (6.4) is minimized in four optimization tests:

- 'Coarse-Vol': coarse mesh with constant volume constraint.
- 'Coarse-Rt': coarse mesh with constant radius (linearized) at leading edge and constant thicknesses (linearized) at two locations, $x/c = 0.05$, and $x/c = 0.35$.
- 'Medium-Vol': medium mesh with constant volume constraint.
- 'Medium-Rt': medium mesh with constant radius (linearized) at leading edge and constant thicknesses (linearized) at two locations, $x/c = 0.05$, and $x/c = 0.35$.

Results are summarized in Table 6.2.

The results obtained for the case 'Medium-Rt' are shown on Figures 6.4-6.7. The history of the optimization is shown in Figure 6.4. Figure 6.5 shows that the wave drag is reduced by 69 drag-counts at design point 1, 96 drag-counts at design point 2, and 1 drag-counts at design point 3, which represents, in relative variations, -53% , -53% , and -4% , respectively. These results do not reflect the weighting of the design points that has been used to build the objective function (6.4). However, the effective scaling, between the different design points, is also determined by the sensitivities of each function (drag, lift, pitch) with respect to the design variables. In particular, the sensitivities of the wave drag vary with the Mach number, that is, larger Mach numbers involve larger sensitivities (as shown in Table 3 in Paper A). The lift and pitch moment coefficients, at design points 1 and 2, are within 1%, of their initial values. The lift and pitch moment coefficients, at design points 3, are within 3% and 2%, respectively, of their initial values. However, during optimization, the constraints are satisfying only after a few iterations.

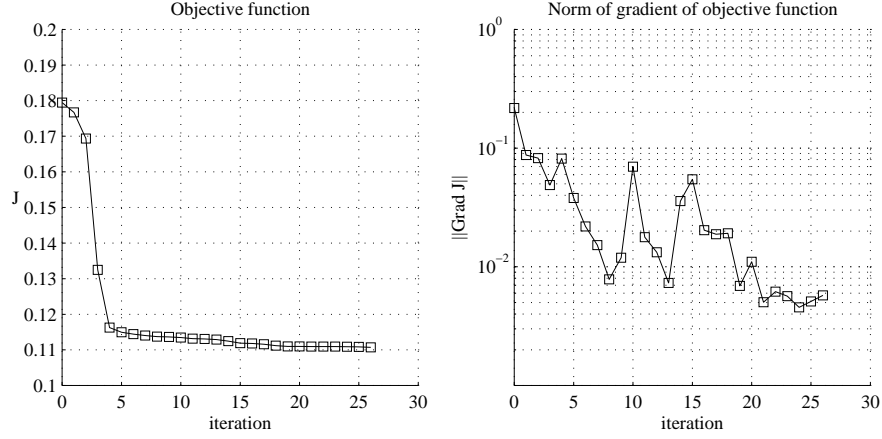


Figure 6.4: Case Medium-Rt: objective function and gradient norm.

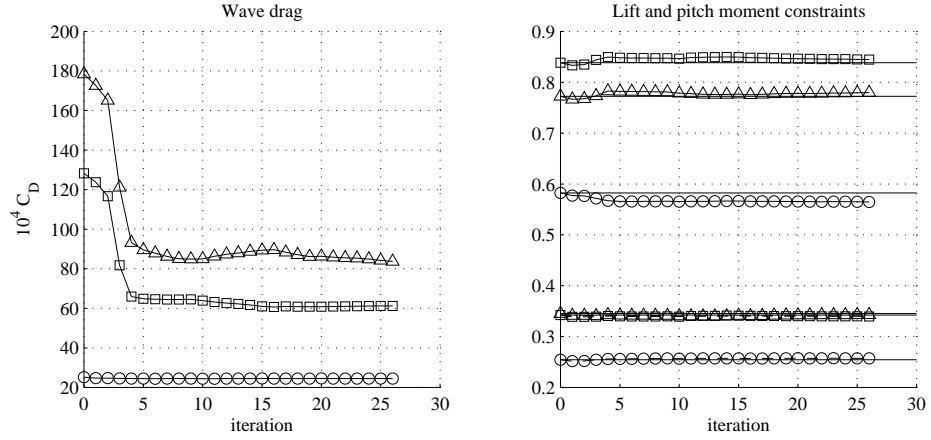


Figure 6.5: Case Medium-Rt: wave drag (left-solid), lift (right-solid) and pitch moment (right-dash) coefficients at design points 1 (square), 2 (triangle) and 3 (circle).

The reduction of the wave drag is achieved (Figure 6.7) by a smoothing of the shock waves at design points 1 and 2. The changes on the geometry of the airfoil (Figure 6.6) are a flattening of the first half of the airfoil, on the upper side, which is balanced by the lower side being thicker, on the first half, due to the constraints on the thicknesses at the positions 5 % and 35 % of the chord length. In contrast, with the changes on the first half of the airfoil, the second half of the wing appears thicker, which is possible since there is no constraints on the volume. Similarly the other applications, the optimization process has not converged to negligible values of $\| \nabla J \|$ (see Figure 6.4) probably due to non exact gradients (see Table 2 in Paper A).

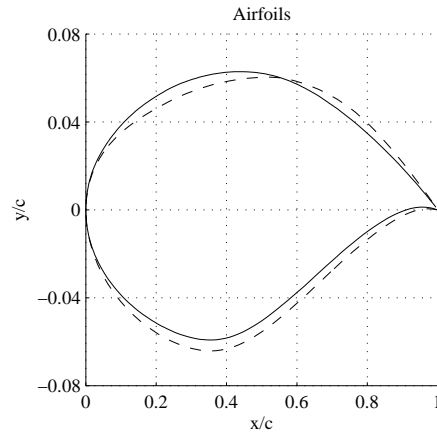


Figure 6.6: Case Medium-Rt: shapes at initial (solid) and final design (dash).

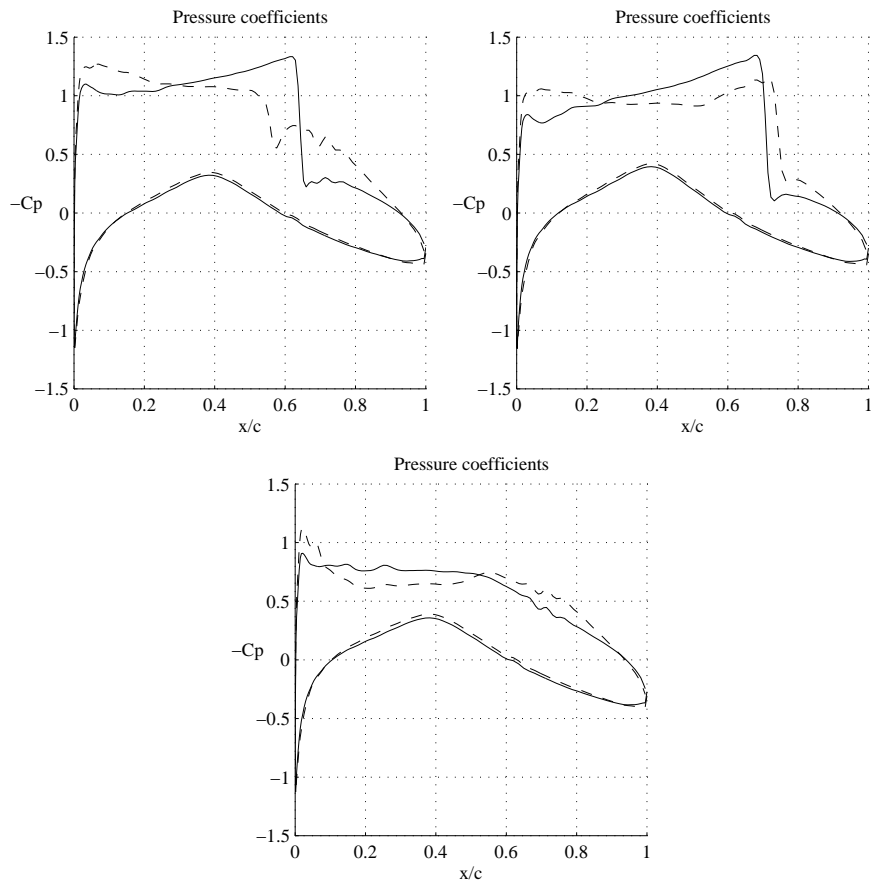


Figure 6.7: Case Medium-Rt: pressure coefficients at initial (solid) and final design (dash), at design points 1 (upper left), 2 (upper right) and 3 (lower).

	C_D	C_L	C_M
1-Coarse (Initial)	1.4×10^{-2}	8.01×10^{-1}	3.28×10^{-1}
1-Coarse-Vol	7.8×10^{-3}	7.95×10^{-1}	3.26×10^{-1}
1-Coarse-Rt	8.7×10^{-3}	8.00×10^{-1}	3.26×10^{-1}
2-Coarse (Initial)	1.8×10^{-2}	7.37×10^{-1}	3.29×10^{-1}
2-Coarse-Vol	9.0×10^{-3}	7.43×10^{-1}	3.30×10^{-1}
2-Coarse-Rt	1.1×10^{-3}	7.40×10^{-1}	3.29×10^{-1}
3-Coarse (Initial)	3.4×10^{-3}	5.60×10^{-1}	2.45×10^{-1}
3-Coarse-Vol	3.2×10^{-3}	5.46×10^{-1}	2.45×10^{-1}
3-Coarse-Rt	3.4×10^{-3}	5.44×10^{-1}	2.46×10^{-1}
1-Medium (Initial)	1.3×10^{-2}	8.38×10^{-1}	3.42×10^{-1}
1-Medium-Vol	5.9×10^{-3}	8.41×10^{-1}	3.40×10^{-1}
1-Medium-Rt	6.1×10^{-3}	8.45×10^{-1}	3.40×10^{-1}
2-Medium (Initial)	1.8×10^{-2}	7.72×10^{-1}	3.45×10^{-1}
2-Medium-Vol	8.6×10^{-3}	7.73×10^{-1}	3.42×10^{-1}
2-Medium-Rt	8.4×10^{-3}	7.79×10^{-1}	3.43×10^{-1}
3-Medium (Initial)	2.5×10^{-3}	5.82×10^{-1}	2.54×10^{-1}
3-Medium-Vol	2.4×10^{-3}	5.67×10^{-1}	2.58×10^{-1}
3-Medium-Rt	2.4×10^{-3}	5.65×10^{-1}	2.57×10^{-1}

Table 6.2: Summary of results of multi-point optimization (the indexes indicate the design point).

6.3 Delay of transition optimization

Here we report the test case T32 in Paper A, which aims at the simultaneous reduction of the wave drag and of the disturbance energy under constraints on the lift and pitch moment coefficients, and under geometric constraints. The conditions are described by $M_1 = 0.734$, $\alpha_1 = 2.19$, and $Re_\infty = 6.5 \times 10^6$ (Reynolds number). The calculations are carried out on the medium mesh previously described.

The objective function is designed to reduce the energy of a certain boundary-layer disturbance together with the wave drag (see § 2.3 in Paper A). This is done by adding measures of the two quantities and weighting them with respect to their value at initial design in order to obtain a balanced reduction of these measures. Furthermore, changes in the lift, and pitch-moment coefficients are penalized. The resulting objective function is

$$J = \lambda_U E_1 + \lambda_D C_D + \frac{1}{2} \lambda_L (C_L - C_L^0)^2 + \frac{1}{2} \lambda_M (C_M - C_M^0)^2, \quad (6.5)$$

where E_1 is a measure of the kinetic energy of a disturbance in the boundary layer above the airfoil. The measure is a streamwise integral over a domain

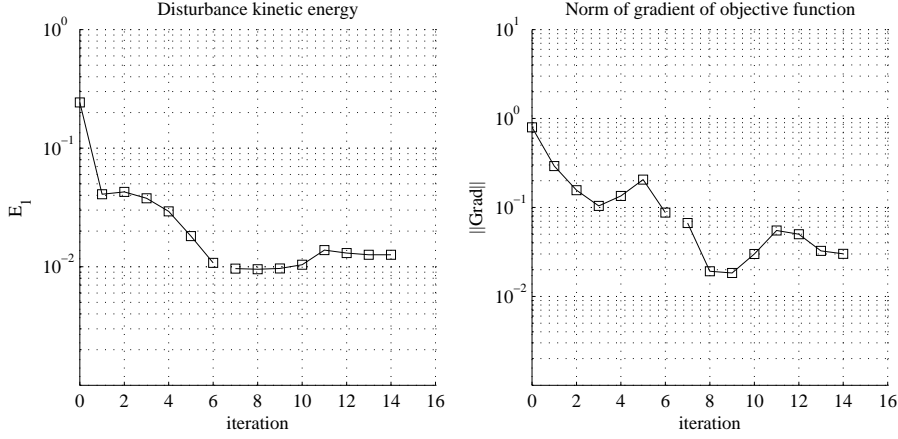


Figure 6.8: T32 - objective function and gradient.

[0.043, 0.45], where each values is a distance from the leading edge, divided by the chord length.

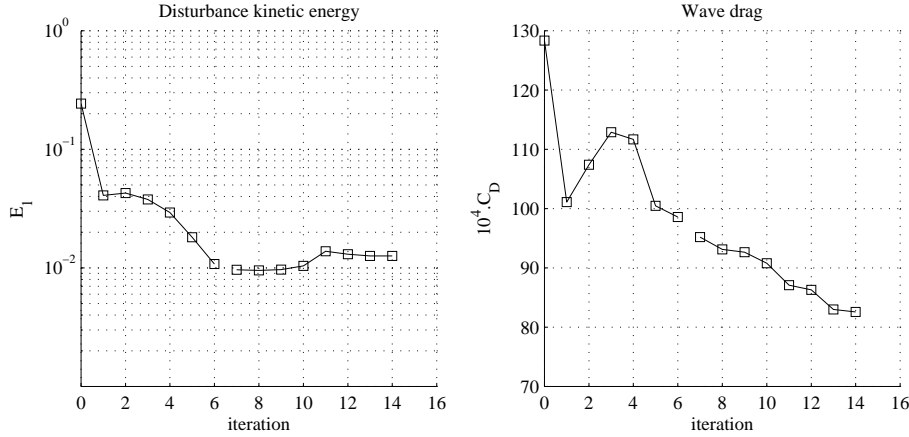


Figure 6.9: Case T32 - disturbance kinetic energy and wave drag.

Moreover, the coupling of the Euler solver (Edge code) to the codes that solve the mean flow in the boundary layer and the propagation of disturbances is limited by assumptions on the position of the stagnation point. Therefore it is necessary, in this case, to fix part of the geometry of the wing in the neighborhood of the leading edge, where the stagnation point is located. In addition to the above geometric constraints, a constraint imposes a constant volume during the optimization, in the same fashion as for the tests of drag reduction that were carried out in §6.1 and §6.2.

The optimization history is given in Figure 6.8–6.10.

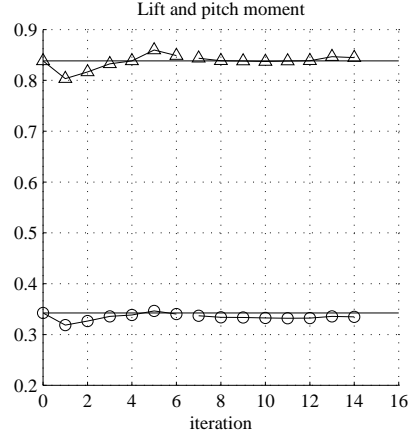


Figure 6.10: Case T32 - Lift (triangle-solid) and pitch moment (circle-solid) coefficients. The values at initial design are indicated at each step (solid).

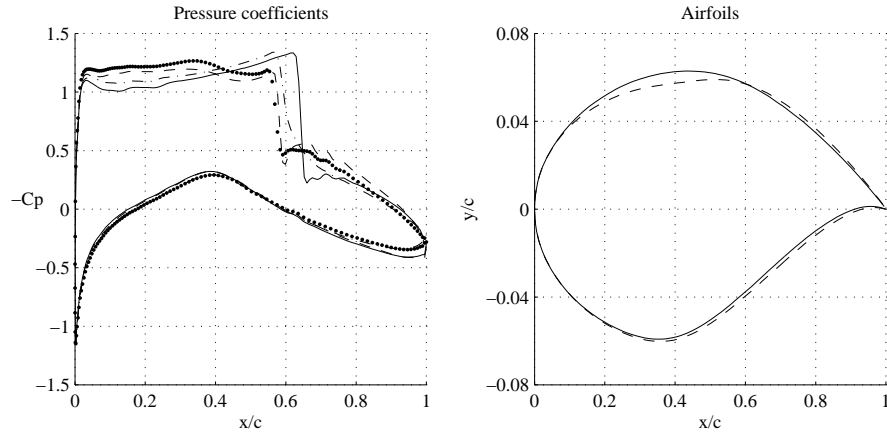


Figure 6.11: Case T32 - pressure coefficients and shapes at initial design for Euler (solid) and RANS (dash-dot), and final design for Euler (dash) and RANS (dot).

In Figure 6.11, a comparison is made between the pressure coefficient and geometry of the initial and final design. Important differences appear in the final design and pressure distribution (Figure 6.11) in comparison with the test of drag optimization Medium-LE (§6.1, Figure 6.3) that has identical geometric constraints but does not include the energy function E_1 in the formulation of the objective function (6.1). In the case T32, the pressure curve is flatter on the first half of the upper side of the airfoil, compared with Medium-LE. The flatter pressure in T32 has a damping effect on the growth of disturbances that are expected to trigger the transition in the boundary-

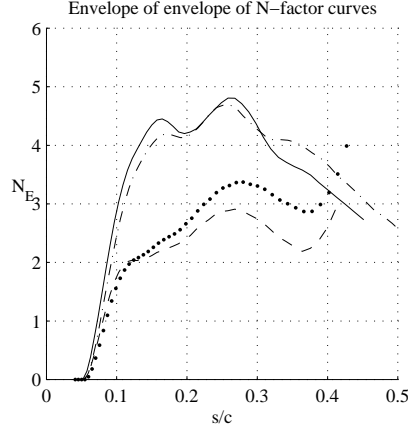


Figure 6.12: Case T32 - envelope of envelopes of N -factor curves. Comparison between initial (solid) and optimal design (dash). A comparison is also made between the initial (dash-dot) and optimal (dot) design, when the pressure distribution is given by the solution of the Reynolds Averaged Navier Stokes equations.

layer (in Paper A, § 5.4). However, a larger reduction of the wave drag is achieved in case Medium-LE (Figure 6.2) than in T32 (Figure 6.9), which is to be expected since the addition of E_1 in the objective function (6.5) acts as an additional constraint on the reduction of the wave drag.

The analysis of the so-called N -factor curves (defined in § 2.2.2 in Paper A), obtained for various modes of disturbances, is common for transition prediction. The Envelope of Envelope (EoE) curves (Figure 6.12) indicate the maxima of the N -factor curves, obtained over all modes that are used for the analysis. Transition is generally assumed to occur at the position where the EoE curve reaches an empirically determined value. In our investigations, the EoE curve is an indicator of the efficiency of the control of the transition by optimization of the shape. The optimization is carried out for one mode, shown in Figure 9 in Paper A, and a post-optimization analysis is therefore required in order to investigate if such an approach is appropriate. As it is shown by the Figure 6.12, all modes have effectively been damped by optimization. Furthermore, an analysis (EoE curve) is carried out by using the pressure distribution from Reynolds Averaged Navier–Stokes calculations¹ (RANS) at initial and final design. Note that the position of the transition needs to be specified for the RANS calculation. Using the curves from the Euler calculations (Figure 6.12), the position was taken as the maximum of the EoE curve at initial design ($s/c = 0.26$). At final design the level of the (Euler) EoE curve is below the level at which transition has

¹The turbulence model used is the EARSM by [36] and the $k - \omega$ model.

been supposed to occur at initial design. Therefore, the point of transition is placed at the downstream position of the computational domain of the BLE. The similarity of the EoE curves, obtained by RANS calculation with the ones obtained by Euler calculation, validates the optimization based on the pressure obtained by inviscid flow calculation.

Chapter 7

Summary and Outlook

An adjoint of the Euler equations, discretized by an unstructured and node-centered finite-volume scheme, has been derived. The system of adjoint equations has been shown to have an edge-based structure similar to that of the Euler equations. The implementation, as well as the derivation of the adjoint equations, have been performed independently of the space dimension.

A method of smoothing has been developed that generates smooth shapes at each design update and that enables us to impose constraints on the geometry. The formulation allows for equality as well as inequality constraints. The latter are reformulated, through this approach, to bound constraints on the parameters of design.

Furthermore, an optimization problem has been formulated with the aim to delay the laminar-to-turbulent transition in the boundary layer. The growth of convectively unstable disturbances have been modeled by successively solving the Euler equations, the boundary layer equations (BLE), and the parabolized stability equations (PSE). Solving successively adjoints of the PSE, the BLE, and the Euler equations has been shown to provide an efficient method for the gradient calculation of the objective function, a measure of the disturbance kinetic energy.

Results have been presented for wave-drag minimization subject to constraints on the lift and pitch moment coefficients, as well as with equality geometric constraints. Preliminary results have been obtained for the delay of the transition by shape optimization. Numerically, they performed as well as the tests on wave drag reduction. Including the modeling of the mechanisms of transition in the optimization has the potential to produce designs that could not have been obtained by classical viscous optimization such as the optimization of the viscous drag based on RANS equations. Indeed,

the RANS equations require empirical data in order to specify the transition point, even if, for this purpose, they make use of a separate stability calculation.

Regarding future work, in order to carry out numerical tests in 3D, it will be required to implement a mesh movement algorithm for general unstructured meshes such as the ones devised in [4], as well as to extend the present smoothing algorithm from curves to surfaces.

With the method of smoothing that has been used, nonlinear geometric constraints could be treated. Tests with inequalities and with nonlinear constraints can be carried out without extensive modifications to the existing implementation.

The accuracy of the gradients should be improved by including all terms in the adjoint discrete artificial dissipation. Furthermore, the derivation of the discrete formulation of the adjoint fluxes related to the fluid viscous dissipation will be carried out, which will allow shape optimization with the RANS equations. This derivation could, in a first approximation, neglect the variations of the turbulence quantities that are used to compute the so-called turbulence viscosity. This term is added to the fluid viscosity when calculating the viscous dissipation. As for the other fluxes, the implementation of the flux in EDGE and the derivation of the discrete adjoint are independent of the dimension.

Although the penalization technique revealed to be an efficient and simple method to impose constraints on the values of the lift and pitch moment coefficients, it would be preferable to specify bound constraints on the lift, for example. Therefore, it would be interesting to carry out tests with an algorithm dedicated to nonlinear constrained optimization, such as a Sequential Quadratic Programming algorithm.

Acknowledgements

I am grateful to my advisor Doctor Martin Berggren for spending so much time and energy trying to share with me his expertise and new ideas about optimal control. Martin has also initiated the collaboration I had with Jan Pralits, and organized everything for a research visit I did at the Computer Science Research Institute (CSRI), USA.

Many thanks to the staff and PhD students at the Division of Scientific Computing (TDB). There I have enjoyed the football, the innebandy, and the internal conferences. I am particularly grateful for the high quality of the lectures and seminars at TDB.

I want to express my gratitude to the staff at the Aeronautics Division, FFA, for technical support as well as for the friendly atmosphere during the long periods I have spent, there, for the development and testing of my programs.

Working together with Jan Pralits has been a great experience; may it happen again.

I want to thank Bart van Bloemen Waanders and Roscoe A. Bartlett, for their technical collaboration during my research visit at CSRI, USA.

Funding from the Parallel and Scientific Computing Institute (PSCI) is greatly acknowledged. Partial support for this research has been obtained through the Department of Energy, USA, for my visit at CSRI.

Thanks to Anita, and our kids, Mikael and Anna, I have not only breathed numbers and drunk coffee, I have also discovered beautiful scandinavian landscapes and re-discovered fairy-tales.

Bibliography

- [1] W.K. Anderson and D.L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.
- [2] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations. In *Special Course on Unstructured Methods for Advection Dominated Flows*, pages 6–1–6–61. AGARD Report 787, May 1991.
- [3] O. Baysal and K. Ghayour. Continuous adjoint sensitivities for optimization with general cost functionals on unstructured meshes. *AIAA Journal*, 39(1):48–55, 2001.
- [4] M. Berggren. Edge-based mesh movement strategies, 2003. Unpublished preprint, Sandia National Laboratories.
- [5] C.H. Bischof, A. Carle, P. Hovland, P. Khademi, and A.XS Mauer. Adifor 2.0 user’s guide (revision d). Technical Report 192, Mathematics and Computer Science Division, Argonne National Laboratory, 1998.
- [6] G. Bugea and E. Oñate. Optimum aerodynamic shape design for fluid flow problems including mesh adaptivity. *Int. J. Numer. Meth. Fluids*, 30:161–178, 1999.
- [7] G.W. Burgreen and O. Baysal. Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis. *AIAA Journal*, 34(9):1761–1170, 1996.
- [8] G.W. Burgreen, O. Baysal, and M.E. Eleshaky. Improving the efficiency of aerodynamic shape optimization. *AIAA Journal*, 32(1):69–76, 1994.
- [9] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. Technical Report NAM-08, Northwestern University, Department of Electrical Engineering and Computer Science, Evanston Il 60208, 1994.
- [10] P.H. Cook, M.A. McDonald, and M.C.P. Firmin. Aerofoil RAE 2822 – pressure distributions, and boundary layer and wake measurements.

- Experimental data base for computer program assessment, AGARD-AR-138, 1979.
- [11] G. Cowles and L. Martinelli. A control-theory based method for shape design in incompressible viscous flow using rans. *AIAA Paper*, 2000.
 - [12] P. Eliasson. Edge, a Navier–Stokes solver, for unstructured grids. Technical Report FOI-R-0298-SE, Swedish Defence Research Agency, Stockholm, November 2001.
 - [13] J. Elliot. *Aerodynamic based on the Euler and Navier–Stokes equations using unstructured grids*. PhD thesis, MIT Dept. of Aero. and Astro., 1998.
 - [14] J. Elliot and J. Peraire. Constrained, multipoint shape optimization for complex 3d configurations. *The Aeronautical Journal*, pages 365–376, August/September 1998. Paper no. 2375.
 - [15] O. Enoksson. Shape optimization in compressible inviscid flow. Licenciata Thesis LiU-TEK-LIC-2000:31, 2000. Institute of Technology, Lindköpings University, Dept. of Math.
 - [16] C. Faure and Y. Ppegay. Odyssée user’s guide version 1.7. Technical Report RT-0224, INRIA, 1998. Webb address: www-sop.inria.fr/tropics/tapenade.html.
 - [17] P.D. Frank and G.R. Shubin. A comparison of optimization-based approaches for a model computational aerodynamics design problem. *Journal of Computational Physics*, 98:74–89, 1992.
 - [18] R. Giering and T. Kaminski. Recipes for adjoint code construction. *ACM Tans. Math. Software*, 24(4):437–474, 1998.
 - [19] M.B. Giles. On the use of runge-kutta time-marching and multigrid for the solution of steady adjoint equations. Technical Report 00/10, Oxford University Computing Laboratory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, England OX1 3QD, 2000.
 - [20] M.B. Giles. Algorithm developments for discrete adjoint methods. Technical Report 01/15, Oxford University Computing Laboratory, Numerical Analysis Group, Wolfson Building, Parks Road, Oxford, England OX1 3QD, 2001.
 - [21] M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Control*, 65:393–415, 2000.
 - [22] A. Griewank, D. Juedes, H. Mitev, J. Utke, O. Vogel, and A. Walther. Adol-c: A package for the automatic differentiation of algorithms written in c/c++. *ACM TOMS*, 22(2):131–167, 1996. Algor. 755.

- [23] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.
- [24] A. Jameson, N.A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier-Stokes equations. *AIAA Paper no 97-0101*, 1997.
- [25] J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.
- [26] B. Mohammadi. A new optimal shape procedure for inviscid and viscous turbulent flows. *Int. J. Numer. Meth. Fluids*, 25:183–203, 1997.
- [27] B. Mohammadi and O. Pironneau. Mesh adaption and automatic differentiation in a cad-free framework for optimal shape design. *Int. J. Numer. Meth. Fluids*, 30:127–136, 1999.
- [28] S.K. Nadarajah and A. Jameson. Studies of the continuous and discrete adjoint approaches to viscous automatic aerodynamic shape optimization. *AIAA Paper No 2001-2530*, 2001.
- [29] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research, 1999.
- [30] O. Pironneau. On optimal profiles in Stokes flow. *J. Fluid Mech.*, 59:117–128, 1973.
- [31] O. Pironneau. *Optimal Shape Design for Elliptic Systems*. Springer Verlag, 1984.
- [32] O. Pironneau and N. Di Césaré. Flow control problem using automatic differentiation in c++. LAN-UPMC report 99013, 1999. Laboratoire d’Analyse Numérique, Université Pierre et Marie Curie, Paris VI, France. Webb address: acm.emath.fr/~dicesare/cpp.php3.
- [33] J. Reuther, J. Alonso, A. Jameson, M. Rimlinger, and J. Saunders. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *J. Aircraft*, 36(1):51–60, 1999.
- [34] B.I. Soemarwoto. *Multi-Point Aerodynamic Design by Optimization*. PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, P.O. Box 5058, 2600 GB Delft, Netherlands, 1996.
- [35] C. Sung and J.H. Kwon. Accurate aerodynamic sensitivity analysis using adjoint equations. *AIAA Journal*, 38(2):243–250, 2000.
- [36] S. Wallin and A. V. Johansson. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. *J. Fluid Mech.*, 403:89–132, 2000.

Appendix A

Median dual grid

The surface vectors of the median dual grid, as defined in [2], are expressed in 2D (§A.1) and in 3D (§A.2) in terms of the nodal coordinates of the primal grid \mathbf{X}_h . These expressions are used in §A.3 to compute gradients of functionals f_X of the nodal coordinates \mathbf{X}_h , when they are defined as $f_X(\mathbf{X}_h) = f_n(\mathbf{n}_h(\mathbf{X}_h))$.

Let \mathcal{N}_i denote the set of nodes that are connected to node i with an edge. The boundary of a control volume V_i can be expressed as the union of control boundaries, that is

$$\partial V_i = \bigcup_{j \in \mathcal{N}_i} \partial V_{ij}, \quad (\text{A.1})$$

where $\partial V_{ij} \equiv \partial V_{ji}$ is the control boundary associated with the edge \vec{ij} , which is defined as

$$\partial V_{ij} = \overline{V}_i \cap \overline{V}_j. \quad (\text{A.2})$$

The surface vector \mathbf{n}_{ij} of the control boundary ∂V_{ij} , associated with edge \vec{ij} , is defined as

$$\mathbf{n}_{ij} = \int_{\partial V_{ij}} \hat{\mathbf{n}} \, dS. \quad (\text{A.3})$$

Note that, by construction,

$$\mathbf{n}_{ji} = -\mathbf{n}_{ij}. \quad (\text{A.4})$$

For a control volume V_i intersecting with the boundary of the domain $\partial\Omega_h$, we will denote by ∂V_{i0} the intersection $\partial\overline{V}_i \cap \partial\Omega_h$, and define the surface vector \mathbf{n}_i as

$$\mathbf{n}_i = \int_{\partial V_{i0}} \hat{\mathbf{n}} \, dS. \quad (\text{A.5})$$

The surface vectors \mathbf{n}_{ij} and \mathbf{n}_i depend on the nodal coordinates \mathbf{X}_h via the definition of the control volumes V_i , as detailed below in §A.1-A.2.

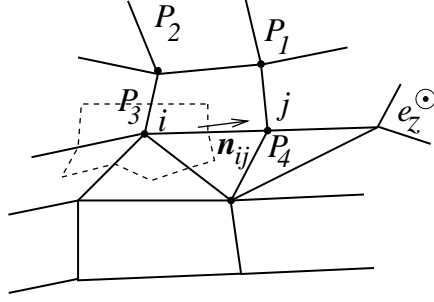


Figure A.1: In 2D the orientation of one vector \mathbf{e}_z , perpendicular to the plane of the mesh, gives the orientation of all the mesh elements. The median dual grid used by Edge is depicted by dashed lines.

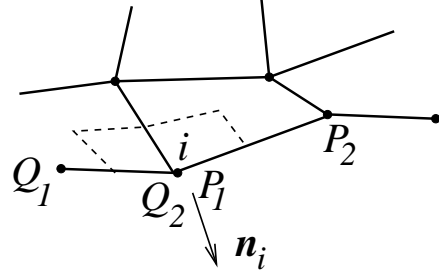


Figure A.2: The boundary edges in 2D are ordered according to the ordering of the interior elements. The dual control volume associated with node i is depicted by dashed lines.

A.1 2D dual grid

Let P denote any 2D mesh element (usually triangles or quadrilaterals) with m_P vertices $\{\mathbf{x}_{k_1}, \dots, \mathbf{x}_{k_{m_P}}\}$, where k_l , $1 \leq l \leq m_P$ are nodal indexes, and \mathbf{x}_{k_l} the d coordinates of node k_l (in 2D, $d = 2$). Let \mathbf{e}_z be a unit vector, perpendicular to the plane, whose direction defines the ordering of P 's vertices (figure A.1). Let $\vec{i}j$ be a directed edge of P . We define

$$\gamma_{\vec{i}j}^P = \begin{cases} 1 & \text{if the orientation of } \vec{i}j \text{ coincides} \\ & \text{with the orientation given by } \mathbf{e}_z, \\ -1 & \text{otherwise.} \end{cases} \quad (\text{A.6})$$

We denote $\bar{\mathbf{x}}^P$ the centroid of the element P and $\bar{\mathbf{x}}_{ij}$ the midpoint of the edge $\vec{i}j$,

$$\begin{aligned} \bar{\mathbf{x}}^P &= \frac{1}{m_P} \sum_{l=1}^{m_P} \mathbf{x}_{k_l}, \\ \bar{\mathbf{x}}_{ij} &= \frac{1}{2} (\mathbf{x}_i + \mathbf{x}_j). \end{aligned} \quad (\text{A.7})$$

Let $\mathcal{E}_{\vec{i}j}$ be the set of elements that admit $\vec{i}j$ as an edge (that is, two elements for an interior edge and one element for an edge on the boundary). The surface vector \mathbf{n}_{ij} (A.3), for this dual mesh [2], is given by

$$\mathbf{n}_{ij} = -\mathbf{e}_z \times \sum_{P \in \mathcal{E}_{\vec{i}j}} \gamma_{\vec{i}j}^P (\bar{\mathbf{x}}^P - \bar{\mathbf{x}}_{ij}). \quad (\text{A.8})$$

The vertices of a boundary edge E are ordered as the vertices of the interior element P to which it belongs (figure A.2). Let \mathcal{E}_i be the set of boundary edges containing node i (that is two). The expression of \mathbf{n}_i , the integrated normal to the dual control face at node i , is

$$\mathbf{n}_i = -\frac{1}{2}\mathbf{e}_z \times \sum_{E \in \mathcal{E}_i} (\mathbf{x}_{k_2} - \mathbf{x}_{k_1}) \quad (\text{A.9})$$

where k_1, k_2 are the vertices of E .

A.2 3D dual grid

Let P be any 3D element (tetrahedron, pyramid, prism, or hexahedron) and denote by S_1 and S_2 the two faces of P sharing the edge $\vec{i}j$. The ordering of the vertices of P is matter of convention and, in the case of the hexahedron, figure A.3 respects the convention used by the preprocessor of EDGE. The faces S_1 and S_2 are oriented according to the direction of edge $\vec{i}j$ (figure A.3). Let $\bar{\mathbf{x}}^P$ be the centroid of P , $\bar{\mathbf{x}}_{ij}$ the mid-point of $\vec{i}j$, and $\bar{\mathbf{x}}^{S_1}$, $\bar{\mathbf{x}}^{S_2}$ the centroids of S_1 and S_2 , respectively. The integrated surface vector associated with edge $\vec{i}j$ will be

$$\mathbf{n}_{ij} = \sum_{P \in \mathcal{E}_{\vec{i}j}} (\bar{\mathbf{x}}^P - \bar{\mathbf{x}}_{ij}) \times (\bar{\mathbf{x}}^{S_2} - \bar{\mathbf{x}}^{S_1}) . \quad (\text{A.10})$$

A similar expression defines the surface vector at the boundary node i , \mathbf{n}_i . Let S be a surface element on the boundary containing node i . Call E_1 and E_2 the two edges of S for which i is a vertex. If E_1 and E_2 are ordered as given by the direction of the outward-directed vector \mathbf{n}_i (in figure A.4), then

$$\mathbf{n}_i = \sum_{S \in \mathcal{E}_i} (\bar{\mathbf{x}}^S - \mathbf{x}_i) \times (\bar{\mathbf{x}}^{E_2} - \bar{\mathbf{x}}^{E_1}) , \quad (\text{A.11})$$

where \mathcal{E}_i is the set of surface elements on the boundary that contain node i .

A.3 Gradients on the primal grid

Consider the set V_n , of all sets of surface vectors \mathbf{n}_h , equipped with the inner product $\langle \cdot, \cdot \rangle_n$, defined as

$$\langle \mathbf{n}_h^1, \mathbf{n}_h^2 \rangle_n = \sum_{\vec{i}j \in \mathcal{E}(\bar{\Omega}_h)} \mathbf{n}_{ij}^1 \cdot \mathbf{n}_{ij}^2 + \sum_{i \in \mathcal{V}(\partial\Omega_h)} \mathbf{n}_i^1 \cdot \mathbf{n}_i^2 \quad \forall \{ \mathbf{n}_h^1, \mathbf{n}_h^2 \} \in (V_n)^2 . \quad (\text{A.12})$$

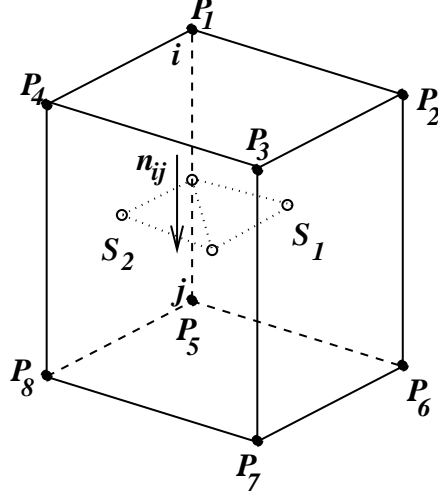


Figure A.3: The dual grid control surface attributed to edge $i\vec{j}$ and contained in element P is depicted by the dotted lines.

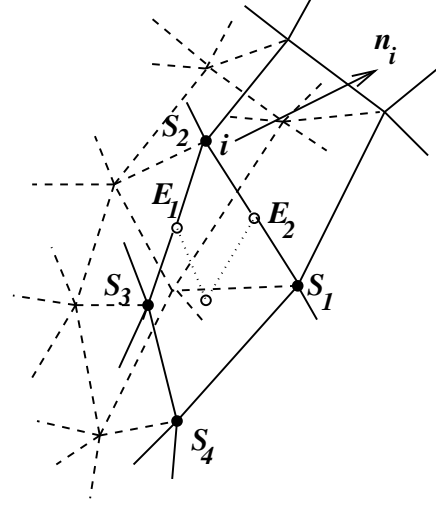


Figure A.4: The dual grid control surface attributed to boundary node i and contained in the surface element S (vertices S_1, S_2, S_3 and S_4) is delimited by the dotted lines.

The gradient, denoted ∇f_n , of a continuously differentiable function f_n , of the variable \mathbf{n}_h , is thus defined as:

$$\delta f_n = \langle \nabla f_n, \delta \mathbf{n}_h \rangle_n \quad \forall \delta \mathbf{n}_h \in V_n, \quad (\text{A.13})$$

where δf_n is the first variation of f_n .

Having defined f_X as above the following relation holds

$$\delta f_X = \delta f_n, \quad (\text{A.14})$$

for $\delta \mathbf{n}_h$ obtained from $\delta \mathbf{X}_h$ through the relations (A.8)-(A.9) or (A.10)-(A.11). Furthermore, given the inner product $\langle \cdot, \cdot \rangle_X$ defined by

$$\langle \mathbf{X}_h^1, \mathbf{X}_h^2 \rangle_X = \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \mathbf{x}_i^1 \cdot \mathbf{x}_i^2 \quad \forall \{\mathbf{X}_h^1, \mathbf{X}_h^2\} \in (V_X)^2 \quad (\text{A.15})$$

where V_X is the set of all nodal displacements \mathbf{X}_h , involves that the relation (A.14) is equivalent to

$$\sum_{i \in \mathcal{V}(\overline{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i = \sum_{\vec{i}\vec{j} \in \mathcal{E}(\overline{\Omega}_h)} (\nabla f_n)_{\vec{i}\vec{j}} \cdot \delta \mathbf{n}_{ij} + \sum_{i \in \mathcal{V}(\partial \Omega_h)} (\nabla f_n)_i \cdot \delta \mathbf{n}_i, \quad (\text{A.16})$$

where, given the mapping $\mathbf{X}_h \rightarrow \mathbf{n}_h$, defined in 2D by (A.8)-(A.9), and in 3D by (A.10)-(A.11), $\delta \mathbf{n}_{ij}$ and $\delta \mathbf{n}_i$ are defined as

$$\delta \mathbf{n}_{ij} = \left\langle \frac{d\mathbf{n}_{ij}}{d\mathbf{X}_h}, \delta \mathbf{X}_h \right\rangle_X \quad \text{and} \quad \delta \mathbf{n}_i = \left\langle \frac{d\mathbf{n}_i}{d\mathbf{X}_h}, \delta \mathbf{X}_h \right\rangle_X \quad \forall \delta \mathbf{X}_h \in V_X. \quad (\text{A.17})$$

A.3.1 Gradients ∇f_X in 2D

For an arbitrary $\delta \mathbf{X}_h \in V_X$, relations (A.8)-(A.9) yield

$$\begin{aligned} \delta \mathbf{n}_{ij} &= -\mathbf{e}_z \times \sum_{P \in \mathcal{E}_{ij}^+} \gamma_{ij}^P (\delta \bar{\mathbf{x}}^P - \delta \bar{\mathbf{x}}_{ij}) , \\ \delta \mathbf{n}_i &= -\frac{1}{2} \mathbf{e}_z \times \sum_{E \in \mathcal{E}_i} (\delta \mathbf{x}_{k_2} - \delta \mathbf{x}_{k_1}) . \end{aligned} \quad (\text{A.18})$$

Making use of (A.18) in (A.16) yields

$$\begin{aligned} \sum_{i \in \mathcal{V}(\bar{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i &= - \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega}_h)} (\nabla f_n)_{\vec{ij}} \cdot \mathbf{e}_z \times \sum_{P \in \mathcal{E}_{ij}^+} \gamma_{ij}^P (\delta \bar{\mathbf{x}}^P - \delta \bar{\mathbf{x}}_{ij}) \\ &\quad - \sum_{i \in \mathcal{V}(\partial \Omega_h)} (\nabla f_n)_i \cdot \frac{1}{2} \mathbf{e}_z \times \sum_{E \in \mathcal{E}_i} (\delta \mathbf{x}_{k_2} - \delta \mathbf{x}_{k_1}) . \end{aligned} \quad (\text{A.19})$$

Using the identities

$$\mathbf{v}_1 \cdot \mathbf{v}_2 \times \mathbf{v}_3 = -\mathbf{v}_3 \cdot \mathbf{v}_2 \times \mathbf{v}_1 = \mathbf{v}_2 \cdot \mathbf{v}_3 \times \mathbf{v}_1 , \quad (\text{A.20})$$

which hold between three vectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ in \mathbb{R}^3 , expression (A.19) may be written

$$\begin{aligned} \sum_{i \in \mathcal{V}(\bar{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i &= \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega}_h)} \sum_{P \in \mathcal{E}_{ij}^+} \gamma_{ij}^P (\delta \bar{\mathbf{x}}^P - \delta \bar{\mathbf{x}}_{ij}) \cdot \mathbf{e}_z \times (\nabla f_n)_{\vec{ij}} \\ &\quad + \sum_{i \in \mathcal{V}(\partial \Omega_h)} \sum_{E \in \mathcal{E}_i} (\delta \mathbf{x}_{k_2} - \delta \mathbf{x}_{k_1}) \cdot \frac{1}{2} \mathbf{e}_z \times (\nabla f_n)_i , \end{aligned} \quad (\text{A.21})$$

which suggests an edge-based algorithm for the calculation of ∇f_X .

An alternative, which is implemented in our programs, is to use an element-based algorithm. Denote by $\mathcal{P}(\Omega_h)$ the set of all 2D elements in Ω_h , $\mathcal{S}(\partial \Omega_h)$ the set of all edge elements on the boundary, and $\mathcal{E}(P)$ the set of the edges

that belong to the element P . Switching the order of summation in expression (A.21) yields

$$\begin{aligned} \sum_{i \in \mathcal{V}(\bar{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i &= \sum_{P \in \mathcal{P}(\Omega_h)} \sum_{\vec{ij} \in \mathcal{E}(P)} (\delta \bar{\mathbf{x}}^P - \delta \bar{\mathbf{x}}_{ij}) \cdot \gamma_{ij}^P \mathbf{e}_z \times (\nabla f_n)_{\vec{ij}} \\ &+ \sum_{E \in \mathcal{S}(\delta \Omega_h)} (\delta \mathbf{x}_{k_2} - \delta \mathbf{x}_{k_1}) \cdot \frac{1}{2} \sum_{i \in E} \mathbf{e}_z \times (\nabla f_n)_i . \end{aligned} \quad (\text{A.22})$$

Making use of the element-based data that defines the primal grid, expression (A.22) is used to build an algorithm for calculating $(\nabla f_X)_i$, for all nodes i in the domain of computation.

A.3.2 Gradients ∇f_X in 3D

For an arbitrary $\delta \mathbf{X}_h \in \mathbf{V}_X$, relations (A.10)-(A.11) yield

$$\begin{aligned} \delta \mathbf{n}_{ij} &= \sum_{P \in \mathcal{E}_{\vec{ij}}} (\delta \bar{\mathbf{x}}^P - \delta \bar{\mathbf{x}}_{ij}) \times (\bar{\mathbf{x}}^{S_2} - \bar{\mathbf{x}}^{S_1}) \\ &+ \sum_{P \in \mathcal{E}_{\vec{ij}}} (\bar{\mathbf{x}}^P - \bar{\mathbf{x}}_{ij}) \times (\delta \bar{\mathbf{x}}^{S_2} - \delta \bar{\mathbf{x}}^{S_1}) , \\ \delta \mathbf{n}_i &= \sum_{S \in \mathcal{E}_i} (\delta \bar{\mathbf{x}}^S - \delta \mathbf{x}_i) \times (\bar{\mathbf{x}}^{E_2} - \bar{\mathbf{x}}^{E_1}) \\ &+ \sum_{S \in \mathcal{E}_i} (\bar{\mathbf{x}}^S - \mathbf{x}_i) \times (\delta \bar{\mathbf{x}}^{E_2} - \delta \bar{\mathbf{x}}^{E_1}) \end{aligned} \quad (\text{A.23})$$

Making use of (A.23) in (A.16) yields

$$\begin{aligned} \sum_{i \in \mathcal{V}(\bar{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i &= \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega}_h)} (\nabla f_n)_{\vec{ij}} \cdot \sum_{P \in \mathcal{E}_{\vec{ij}}} (\delta \bar{\mathbf{x}}^P - \delta \bar{\mathbf{x}}_{ij}) \times (\bar{\mathbf{x}}^{S_2} - \bar{\mathbf{x}}^{S_1}) \\ &+ \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega}_h)} (\nabla f_n)_{\vec{ij}} \cdot \sum_{P \in \mathcal{E}_{\vec{ij}}} (\bar{\mathbf{x}}^P - \bar{\mathbf{x}}_{ij}) \times (\delta \bar{\mathbf{x}}^{S_2} - \delta \bar{\mathbf{x}}^{S_1}) \\ &+ \sum_{i \in \mathcal{V}(\partial \Omega_h)} (\nabla f_n)_i \cdot \sum_{S \in \mathcal{E}_i} (\delta \bar{\mathbf{x}}^S - \delta \mathbf{x}_i) \times (\bar{\mathbf{x}}^{E_2} - \bar{\mathbf{x}}^{E_1}) \\ &+ \sum_{i \in \mathcal{V}(\partial \Omega_h)} (\nabla f_n)_i \cdot \sum_{S \in \mathcal{E}_i} (\bar{\mathbf{x}}^S - \mathbf{x}_i) \times (\delta \bar{\mathbf{x}}^{E_2} - \delta \bar{\mathbf{x}}^{E_1}) . \end{aligned} \quad (\text{A.24})$$

Using the identities (A.20), the relation (A.24) is rewritten as

$$\begin{aligned}
\sum_{i \in \mathcal{V}(\overline{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i &= \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega}_h)} \sum_{P \in \mathcal{E}_{\vec{ij}}} (\delta \overline{\mathbf{x}}^P - \delta \overline{\mathbf{x}}_{ij}) \cdot (\overline{\mathbf{x}}^{S_2} - \overline{\mathbf{x}}^{S_1}) \times (\nabla f_n)_{\vec{ij}} \\
&\quad - \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega}_h)} \sum_{P \in \mathcal{E}_{\vec{ij}}} (\delta \overline{\mathbf{x}}^{S_2} - \delta \overline{\mathbf{x}}^{S_1}) \cdot (\overline{\mathbf{x}}^P - \overline{\mathbf{x}}_{ij}) \times (\nabla f_n)_{\vec{ij}} \\
&\quad + \sum_{i \in \mathcal{V}(\partial \Omega_h)} \sum_{S \in \mathcal{E}_i} (\delta \overline{\mathbf{x}}^S - \delta \mathbf{x}_i) \cdot (\overline{\mathbf{x}}^{E_2} - \overline{\mathbf{x}}^{E_1}) \times (\nabla f_n)_i \\
&\quad - \sum_{i \in \mathcal{V}(\partial \Omega_h)} \sum_{S \in \mathcal{E}_i} (\delta \overline{\mathbf{x}}^{E_2} - \delta \overline{\mathbf{x}}^{E_1}) \cdot (\overline{\mathbf{x}}^S - \mathbf{x}_i) \times (\nabla f_n)_i,
\end{aligned} \tag{A.25}$$

which defines an edge-based algorithm for the calculation of ∇f_X . The equivalent element-based algorithm is obtained making use of the definitions of the sets $\mathcal{P}(\Omega_h)$, $\mathcal{S}(\delta \Omega_h)$ and $\mathcal{E}(P)$, see §A.3.1, which yields

$$\begin{aligned}
\sum_{i \in \mathcal{V}(\overline{\Omega})} (\nabla f_X)_i \cdot \delta \mathbf{x}_i &= \sum_{P \in \mathcal{P}(\Omega_h)} \sum_{\vec{ij} \in \mathcal{E}(P)} (\delta \overline{\mathbf{x}}^P - \delta \overline{\mathbf{x}}_{ij}) \cdot (\overline{\mathbf{x}}^{S_2} - \overline{\mathbf{x}}^{S_1}) \times (\nabla f_n)_{\vec{ij}} \\
&\quad - \sum_{P \in \mathcal{P}(\Omega_h)} \sum_{\vec{ij} \in \mathcal{E}(P)} (\delta \overline{\mathbf{x}}^{S_2} - \delta \overline{\mathbf{x}}^{S_1}) \cdot (\overline{\mathbf{x}}^P - \overline{\mathbf{x}}_{ij}) \times (\nabla f_n)_{\vec{ij}} \\
&\quad + \sum_{S \in \mathcal{S}(\delta \Omega_h)} \sum_{i \in S} (\delta \overline{\mathbf{x}}^S - \delta \mathbf{x}_i) \cdot (\overline{\mathbf{x}}^{E_2} - \overline{\mathbf{x}}^{E_1}) \times (\nabla f_n)_i \\
&\quad - \sum_{i \in \mathcal{V}(\partial \Omega_h)} \sum_{S \in \mathcal{E}_i} (\delta \overline{\mathbf{x}}^{E_2} - \delta \overline{\mathbf{x}}^{E_1}) \cdot (\overline{\mathbf{x}}^S - \mathbf{x}_i) \times (\nabla f_n)_i.
\end{aligned} \tag{A.26}$$

Making use of the element-based data that defines the primal grid, expression (A.26) is used to build an algorithm for calculating $(\nabla f_X)_i$, for all nodes i in the domain of computation.

Appendix B

Discrete sensitivities

B.1 Inner product and differentiation

The adjoint of a linear operator is defined with respect to an inner product. We denote V_w the vector space of all 3-by-1 matrices of tensors of dimension as the conservative variables (3.8). If \mathbf{z} and \mathbf{q} are in V_w , and \mathbf{z}_i and \mathbf{q}_i are their components associated with a mesh point i , we define the scalar $\mathbf{z}_i^T \mathbf{q}_i$ by the usual matrix operation, complemented with a dot product for the second element. Moreover, by summing over all nodes $\mathcal{V}(\overline{\Omega}_h)$, we define the inner product

$$\langle \mathbf{z}, \mathbf{q} \rangle_w = \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \mathbf{z}_i^T \mathbf{q}_i. \quad (\text{B.1})$$

That is, if \mathcal{A} is a linear operator, in $\mathcal{L}(V_w, V_w)$, its adjoint, denoted \mathcal{A}^* is defined by

$$\langle \mathbf{q}, \mathcal{A}\mathbf{z} \rangle_w = \langle \mathcal{A}^* \mathbf{q}, \mathbf{z} \rangle_w, \quad (\text{B.2})$$

for all \mathbf{z} and \mathbf{q} in V_w .

In the next section, the flow equations are differentiated to obtain $\partial \mathcal{A}_h / \partial \mathbf{w}_h$. Further notations that are used is $(\partial \mathcal{A}_h / \partial \mathbf{w}_h)^*$ and $(\partial \mathcal{A}_h / \partial \mathbf{w}_h)^{-1}$, which stand for the adjoint and the inverse of the linear operator $\partial \mathcal{A}_h / \partial \mathbf{w}_h$, respectively. An expression like $(\partial \mathcal{A}_h / \partial \mathbf{w}_h) \delta \mathbf{w}_h$ means the application of the linear operator to the variable $\delta \mathbf{w}_h$.

The flux functions \mathbf{f} (3.7), are 3-by-1 matrices of first and second-order tensors. Various operations need to be defined on \mathbf{f} . As we already have seen in chapter 3 (§3.1), the dot product of a flux function \mathbf{f} with any first-order tensor \mathbf{n} yields 3-by-1 matrices $\mathbf{f} \cdot \mathbf{n}$ (3.11) of the same dimension as the matrix of conservative variables (3.8). These objects may be differentiated with respect to the conservative variables or, with respect to the primitive variables.

For \mathbf{f} being defined by (3.7), the differentiation of $\mathbf{f} \cdot \mathbf{n}$ with respect to the conservative variable \mathbf{w} (3.8) is noted

$$\frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{w}} = \left[\frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \rho}, \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{m}}, \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial E} \right], \quad (\text{B.3})$$

where

$$\begin{aligned} \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \rho} &= \begin{pmatrix} 0 \\ -\mathbf{u} \cdot \mathbf{n} \mathbf{u} + \mathbf{n}(\gamma - 1) \frac{|\mathbf{u}|^2}{2} \\ \mathbf{n} \cdot \mathbf{u} (-\gamma E / \rho + (\gamma - 1) |\mathbf{u}|^2) \end{pmatrix}, \\ \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{m}} &= \begin{pmatrix} \mathbf{n} \\ \mathbf{u} \cdot \mathbf{n} \mathbf{I} + \mathbf{u} \otimes \mathbf{n} - (\gamma - 1) \mathbf{n} \otimes \mathbf{u} \\ \gamma \mathbf{n} E / \rho - (\gamma - 1) \left(\mathbf{n} \cdot \mathbf{u} \mathbf{u} + \frac{|\mathbf{u}|^2}{2} \mathbf{n} \right) \end{pmatrix}, \\ \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial E} &= \begin{pmatrix} 0 \\ (\gamma - 1) \mathbf{n} \\ \gamma \mathbf{u} \cdot \mathbf{n} \end{pmatrix}. \end{aligned} \quad (\text{B.4})$$

The differentiation of the same quantity with respect to the primitive variables \mathbf{v} (3.10) is needed for the differentiation of the flux at a farfield boundary. The operation is denoted

$$\frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{v}} = \left[\frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \rho}, \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{u}}, \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial p} \right], \quad (\text{B.5})$$

where

$$\begin{aligned} \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \rho} &= \begin{pmatrix} \mathbf{u} \cdot \mathbf{n} \\ \mathbf{u} \cdot \mathbf{n} \mathbf{u} \\ \frac{1}{2} |\mathbf{u}|^2 \mathbf{u} \cdot \mathbf{n} \end{pmatrix}, \\ \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{u}} &= \begin{pmatrix} \rho \mathbf{n} \\ \rho \mathbf{u} \cdot \mathbf{n} \mathbf{I} + \rho \mathbf{u} \otimes \mathbf{n} \\ \rho \mathbf{u} \cdot \mathbf{n} \mathbf{u} + \frac{1}{2} \rho |\mathbf{u}|^2 \mathbf{n} + \gamma p \mathbf{n} \end{pmatrix}, \\ \frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial p} &= \begin{pmatrix} 0 \\ \mathbf{n} \\ \gamma \mathbf{u} \cdot \mathbf{n} \end{pmatrix}. \end{aligned} \quad (\text{B.6})$$

The differentiation of dot product $\mathbf{f}^{\text{bc}} \cdot \mathbf{n}$, for instance at a wall (3.17), is denoted

$$\frac{\partial(\mathbf{f}^{\text{bc}} \cdot \mathbf{n})}{\partial \mathbf{w}} = (\gamma - 1) \begin{bmatrix} 0 & \mathbf{0} & 0 \\ \frac{1}{2} |\mathbf{u}|^2 \mathbf{n} & -\mathbf{n} \otimes \mathbf{u} & \mathbf{n} \\ 0 & \mathbf{0} & 0 \end{bmatrix}, \quad (\text{B.7})$$

Dealing with the flux \mathbf{f}^{bc} at a farfield boundary (3.18) we define the differentiation with respect to the conservative variable by freezing the eigenvalues,

denoted λ , prior to the differentiation. This simplification, discussed in §B.3, yields

$$\frac{\partial (\mathbf{f}^{\text{bc}} \cdot \mathbf{n})}{\partial \mathbf{w}} = \frac{\partial (\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{v}^c} \mathbf{L}(\hat{\mathbf{n}}, \mathbf{v}_\infty) \mathbf{H}(\lambda) \mathbf{L}^{-1}(\hat{\mathbf{n}}, \mathbf{v}_\infty) \frac{d\mathbf{v}}{d\mathbf{w}}, \quad (\text{B.8})$$

where $\partial (\mathbf{f} \cdot \mathbf{n}) / \partial \mathbf{v}^c$ denotes the Jacobian with respect to the primitive variables \mathbf{v}^c (3.18). It is obtained by evaluating (B.5) at $\mathbf{v} = \mathbf{v}^c$. The Jacobian $d\mathbf{v}/d\mathbf{w}$ of the transformation conservative-to-primitive variables is

$$\frac{d\mathbf{v}}{d\mathbf{w}} = \begin{bmatrix} 1 & \mathbf{0}^T & 0 \\ -\mathbf{u}/\rho & \mathbf{I}/\rho & 0 \\ (\gamma - 1) \mathbf{u}^2/2 & (1 - \gamma) \mathbf{u}^T & (\gamma - 1) \end{bmatrix}. \quad (\text{B.9})$$

Note that expressions (B.3)-(B.9) are invariant with respect to coordinate system and space dimension.

The “ordinary” Jacobian matrices are obtained by successively choosing \mathbf{n} , in (B.3)-(B.8), as the unit vectors in the Cartesian coordinate directions. For the derivation of the adjoint (in §B.2), \mathbf{n} are replaced by the surface vectors associated with either the edges or the boundary nodes (see appendix A).

Similar operations need to be defined for the fluxes of artificial dissipation. Note that the stencil of the linearized second order artificial dissipation, in the residual at node i (3.21), involves all neighboring nodes of i and all neighbor-to-neighbor nodes. From (3.24) we can define Jacobian matrices of the 2nd order artificial dissipation fluxes as

$$\begin{aligned} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_k} &= \delta_{ki} \epsilon_2^{ij} \mathbf{I} + (\mathbf{w}_i - \mathbf{w}_j) \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{w}_k} \quad \forall k \in \mathcal{N}_j, \\ \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} &= (\mathbf{w}_i - \mathbf{w}_j) \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{w}_l} - \delta_{lj} \epsilon_2^{ij} \mathbf{I} \quad \forall l \in \mathcal{N}_i, \end{aligned} \quad (\text{B.10})$$

where \mathbf{I} is the identity matrix, δ_{ik} is 1 if $k = i$ and 0 otherwise, and, $\partial \epsilon_2^{ij} / \partial \mathbf{w}_k$ is the 1-by-3 matrix of tensors

$$\frac{\partial \epsilon_2^{ij}}{\partial \mathbf{w}_k} = \left[\frac{\partial \epsilon_2^{ij}}{\partial \rho_k}, \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{m}_k}, \frac{\partial \epsilon_2^{ij}}{\partial E_k} \right], \quad (\text{B.11})$$

of which a detailed expression is given in §B.3.2.

For the differentiation of the fourth order artificial dissipation we may use the variation of the undivided Laplace (3.23), that is, $\delta \nabla^2 \mathbf{w}_i$, equal to $\nabla^2 \delta \mathbf{w}_i$, in the sense

$$\nabla^2 \delta \mathbf{w}_i = \sum_{k \in \mathcal{N}_i} (\delta \mathbf{w}_k - \delta \mathbf{w}_i) \quad \forall i \in \mathcal{V}(\bar{\Omega}_h), \quad (\text{B.12})$$

or, the following expressions or the Jacobian matrices, which makes use of relation (B.12)

$$\begin{aligned}\frac{\partial \mathbf{d}_{ij}^{\text{4th}}}{\partial \mathbf{w}_k} &= \epsilon_4^i \mathbf{I} + \epsilon_4^j \delta_{kj} \sum_{l \in \mathcal{N}_j} \mathbf{I} \quad \forall k \in \mathcal{N}_i \\ \frac{\partial \mathbf{d}_{ij}^{\text{4th}}}{\partial \mathbf{w}_l} &= -\epsilon_4^i \delta_{li} \sum_{k \in \mathcal{N}_i} \mathbf{I} - \epsilon_4^j \mathbf{I} \quad \forall l \in \mathcal{N}_j\end{aligned}\tag{B.13}$$

The Jacobians above are 3-by-3 matrices in which the elements are zeroth-, first-, and second-order tensors. If $\mathbf{q} = (q_1, \mathbf{q}_2, q_3)^T$ is a 3-by-1 matrix containing tensors of dimensions like the matrix of conservative variables (3.8), we may define the matrix operations

$$\mathbf{q}^T \left[\frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{w}} \right], \quad \left[\frac{\partial(\mathbf{f} \cdot \mathbf{n})}{\partial \mathbf{w}} \right]^T \mathbf{q},\tag{B.14}$$

in which the component-wise multiplication also includes a dot product for the second element of \mathbf{q} . Moreover, the operation

$$\mathbf{q}^T \mathbf{f} = \mathbf{f}^T \mathbf{q},\tag{B.15}$$

returning a first-order tensor, may also be defined unambiguously. A dot product is involved when multiplying the second component of \mathbf{f} and \mathbf{q} , as usual.

The flux of the second order artificial viscosity is not easily differentiated with respect to the set of surface vectors \mathbf{n}_h . This task is carried out in details in §B.3.3. Meanwhile, based on the definition of ϵ_2^{ij} (3.25), we express the variation of ϵ_2^{ij} for an arbitrary variation of the set of surface vectors $\delta \mathbf{n}_h$:

For all $\vec{ij} \in \mathcal{V}(\Omega_h)$:

$$\begin{aligned}\left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n &= \sum_{k \in \mathcal{N}_j} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{kj}} \cdot \delta \mathbf{n}_{kj} + \sum_{l \in \mathcal{N}_i} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{il}} \cdot \delta \mathbf{n}_{il} \\ &\quad - \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{ij}} \cdot \delta \mathbf{n}_{ij},\end{aligned}\tag{B.16}$$

For all i on the boundary $\mathcal{V}(\partial\Omega_h)$ and j in the interior of the domain $\mathcal{V}(\Omega_h)$:

$$\begin{aligned}\left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n &= \sum_{k \in \mathcal{N}_j} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{kj}} \cdot \delta \mathbf{n}_{kj} + \sum_{l \in \mathcal{N}_i} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{il}} \cdot \delta \mathbf{n}_{il} \\ &\quad - \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{ij}} \cdot \delta \mathbf{n}_{ij} + \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_i} \cdot \delta \mathbf{n}_i,\end{aligned}\tag{B.17}$$

For all \vec{ij} on the boundary:

$$\begin{aligned} \left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n &= \sum_{k \in \mathcal{N}_j} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{kj}} \cdot \delta \mathbf{n}_{kj} + \sum_{l \in \mathcal{N}_i} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{il}} \cdot \delta \mathbf{n}_{il} \\ &\quad - \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{ij}} \cdot \delta \mathbf{n}_{ij} + \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_i} \cdot \delta \mathbf{n}_i + \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_j} \cdot \delta \mathbf{n}_j, \end{aligned} \quad (\text{B.18})$$

B.2 Sensitivity calculations

The variation of a function $f(\mathbf{w}_h, \mathbf{n}_h)$, where \mathbf{w}_h is solution of the state equation (2.3) is

$$\delta f = \left\langle \frac{\partial f}{\partial \mathbf{w}_n}, \delta \mathbf{w}_h \right\rangle_w + \left\langle \frac{\partial f}{\partial \mathbf{n}_n}, \delta \mathbf{n}_h \right\rangle_n, \quad (\text{B.19})$$

for an arbitrary variation $\delta \mathbf{w}_h$ of the state \mathbf{w}_h , and for $\delta \mathbf{n}_h$ related to the variation of the nodal coordinates $\delta \mathbf{X}_h$, as defined by (A.8)-(A.9), in 2D, and, (A.10)-(A.11), in 3D. In the following we express:

- 1 the *sensitivity equations* of the state equation, that is (2.3) linearized with respect to the flow state \mathbf{w}_h and the surface vectors \mathbf{n}_h ,

$$\frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \delta \mathbf{w}_h = - \frac{\partial \mathcal{A}_h}{\partial \mathbf{n}_h} \delta \mathbf{n}_h, \quad (\text{B.20})$$

- 2 the inner product of an arbitrary variable \mathbf{z}_h with the *sensitivity equations* of the flow state, that is

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w = - \left\langle \mathbf{z}_h, \frac{\partial \mathcal{A}_h}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w, \quad (\text{B.21})$$

- 3 expression for the adjoint (transposed) operator defined by (B.2) in which \mathcal{A} is replaced by $\partial \mathcal{A}_h / \partial \mathbf{w}_h$,
- 4 definition of an adjoint state \mathbf{w}_h^* ,
- 5 and finally, that if the variation of the flow state $\delta \mathbf{w}_h$ is the solution of the sensitivity equations, the variation of a function f , with respect to $\delta \mathbf{w}_h$ can be expressed with respect to the adjoint state \mathbf{w}_h^* .

Splitting the residuals (3.21) will clarify the derivation of the adjoint equations. The convective residuals are denoted \mathcal{P} , the residuals of the fluxes at the boundaries \mathcal{B} , the residuals of the second order artificial dissipation

fluxes \mathcal{D}_2 , and the residuals of the fourth order artificial dissipation fluxes \mathcal{D}_4 . Thus, the residuals (3.21) can be written

$$\mathcal{A}_h(\mathbf{w}_h, \mathbf{n}_h) \equiv \mathcal{P}(\mathbf{w}_h, \mathbf{n}_h) + \mathcal{B}(\mathbf{w}_h, \mathbf{n}_h) + \mathcal{D}_2(\mathbf{w}_h, \mathbf{n}_h) + \mathcal{D}_4(\mathbf{w}_h, \mathbf{n}_h), \quad (\text{B.22})$$

where

$$\begin{aligned} \mathcal{P}(\mathbf{w}_h, \mathbf{n}_h)_i &= \sum_{j \in \mathcal{N}_i} \mathbf{f}_{ij} \cdot \mathbf{n}_{ij} & \forall i \in \mathcal{V}(\bar{\Omega}_h), \\ \mathcal{B}(\mathbf{w}_h, \mathbf{n}_h)_i &= \mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i & \forall i \in \partial\mathcal{V}(\Omega_h), \\ \mathcal{B}(\mathbf{w}_h, \mathbf{n}_h)_i &= \mathbf{0} & \forall i \in \mathcal{V}(\Omega_h), \\ \mathcal{D}_2(\mathbf{w}_h, \mathbf{n}_h)_i &= \sum_{j \in \mathcal{N}_i} \mathbf{d}_{ij}^{\text{2nd}} & \forall i \in \mathcal{V}(\bar{\Omega}_h), \\ \mathcal{D}_4(\mathbf{w}_h, \mathbf{n}_h)_i &= \sum_{j \in \mathcal{N}_i} \mathbf{d}_{ij}^{\text{4th}} & \forall i \in \mathcal{V}(\bar{\Omega}_h). \end{aligned} \quad (\text{B.23})$$

Differentiating equation (2.3), using the splitting (B.22), gives the equations (B.20) of the variations of the fluid state $\delta\mathbf{w}_h$, also called *flow sensitivities*,

$$\begin{aligned} \frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \delta\mathbf{w}_h + \frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \delta\mathbf{w}_h + \\ \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \delta\mathbf{w}_h + \frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \delta\mathbf{w}_h = -\frac{\partial \mathcal{P}}{\partial \mathbf{n}_h} \delta\mathbf{n}_h - \frac{\partial \mathcal{B}}{\partial \mathbf{n}_h} \delta\mathbf{n}_h \\ - \frac{\partial \mathcal{D}_2}{\partial \mathbf{n}_h} \delta\mathbf{n}_h - \frac{\partial \mathcal{D}_4}{\partial \mathbf{n}_h} \delta\mathbf{n}_h. \end{aligned} \quad (\text{B.24})$$

By (B.23), each term in (B.24) is detailed below

$$\begin{aligned} \left(\frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \delta\mathbf{w}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \delta\mathbf{w}_i + \frac{\partial(\mathbf{f}_j \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_j} \delta\mathbf{w}_j \right] & \forall i \in \mathcal{V}(\bar{\Omega}_h), \\ \left(\frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \delta\mathbf{w}_h \right)_i &= \frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \delta\mathbf{w}_i & \forall i \in \mathcal{V}(\partial\Omega_h), \\ \left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \delta\mathbf{w}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \left[\sum_{l \in \mathcal{N}_i} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} \delta\mathbf{w}_l + \sum_{k \in \mathcal{N}_j} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_k} \delta\mathbf{w}_k \right] & \forall i \in \mathcal{V}(\bar{\Omega}_h), \\ \left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \delta\mathbf{w}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \left[\sum_{k \in \mathcal{N}_i} \frac{\partial \mathbf{d}_{ij}^{\text{4th}}}{\partial \mathbf{w}_k} \delta\mathbf{w}_k + \sum_{l \in \mathcal{N}_j} \frac{\partial \mathbf{d}_{ij}^{\text{4th}}}{\partial \mathbf{w}_l} \delta\mathbf{w}_l \right] & \forall i \in \mathcal{V}(\bar{\Omega}_h), \end{aligned} \quad (\text{B.25})$$

and

$$\begin{aligned}
\left(\frac{\partial \mathcal{P}}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \mathbf{f}_{ij} \cdot \delta \mathbf{n}_{ij} & \forall i \in \mathcal{V}(\overline{\Omega}_h), \\
\left(\frac{\partial \mathcal{B}}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right)_i &= \mathbf{f}_i^{\text{bc}} \cdot \delta \mathbf{n}_i & \forall i \in \mathcal{V}(\partial \Omega_h), \\
\left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n (\mathbf{w}_i - \mathbf{w}_j) & \forall i \in \mathcal{V}(\overline{\Omega}_h), \\
\left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right)_i &= \mathbf{0} & \forall i \in \mathcal{V}(\overline{\Omega}_h),
\end{aligned} \tag{B.26}$$

where $\left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n$ is detailed in §B.3.3. The left side of equation (B.24) is linear in $\delta \mathbf{w}_h$, that is, linear in each component $\delta \mathbf{w}_i$, as shown in (B.25) and thus defines the action of the Jacobian of the discrete Euler residuals, that is, the operator $\partial \mathcal{A}_h / \partial \mathbf{w}_h$ in $\mathcal{L}(V_w, V_w)$, on the variable $\delta \mathbf{w}_h$.

B.2.1 Adjoint operator

Let \mathbf{z}_h be an arbitrary element in V_w . Multiplying equations (B.24) from the left with \mathbf{z}_h , in the sense of the inner product (B.1), yields

$$\begin{aligned}
\left\langle \mathbf{z}, \frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w + \left\langle \mathbf{z}, \frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w + \\
\left\langle \mathbf{z}, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w + \left\langle \mathbf{z}, \frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w &= - \left\langle \mathbf{z}, \frac{\partial \mathcal{P}}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w - \left\langle \mathbf{z}, \frac{\partial \mathcal{B}}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w \\
&\quad - \left\langle \mathbf{z}, \frac{\partial \mathcal{D}_2}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w - \left\langle \mathbf{z}, \frac{\partial \mathcal{D}_4}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w,
\end{aligned} \tag{B.27}$$

where

$$\begin{aligned}
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left[\mathbf{z}_i^T \frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \delta \mathbf{w}_i + \mathbf{z}_i^T \frac{\partial(\mathbf{f}_j \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_j} \delta \mathbf{w}_j \right] \\
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{z}_i^T \frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \delta \mathbf{w}_i \\
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \left[\mathbf{z}_i^T \sum_{l \in \mathcal{N}_i} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} \delta \mathbf{w}_l + \mathbf{z}_i^T \sum_{k \in \mathcal{N}_j} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_k} \delta \mathbf{w}_k \right] \\
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \left[\mathbf{z}_i^T \sum_{l \in \mathcal{N}_i} \frac{\partial \mathbf{d}_{ij}^{\text{4th}}}{\partial \mathbf{w}_l} \delta \mathbf{w}_l + \mathbf{z}_i^T \sum_{k \in \mathcal{N}_j} \frac{\partial \mathbf{d}_{ij}^{\text{4th}}}{\partial \mathbf{w}_k} \delta \mathbf{w}_k \right]
\end{aligned} \tag{B.28}$$

by (B.25), and

$$\begin{aligned}
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{P}}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \mathbf{z}_i^T \mathbf{f}_{ij} \cdot \delta \mathbf{n}_{ij} \\
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{B}}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{z}_i^T \mathbf{f}_i^{\text{bc}} \cdot \delta \mathbf{n}_i \\
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w &= \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \mathbf{z}_i^T \left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n (\mathbf{w}_i - \mathbf{w}_j) \\
\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_4}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w &= \mathbf{0}
\end{aligned} \tag{B.29}$$

by (B.26)

To proceed in the identification of the adjoint operator $(\partial \mathcal{A}_h / \partial \mathbf{w}_h)^*$, we need to apply the linearized operator $\partial \mathcal{A}_h / \partial \mathbf{w}_h$ on an arbitrary variable. We denote \mathbf{q}_h an arbitrary element of V_w .

Each right side of (B.28) is a double sum of the form

$$\sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij},$$

in which \mathbf{e}_{ij} is an algebraic expression associated with edge $\vec{i}j$. Contributions to the double sum associated with edge $\vec{i}j$ is obtained precisely twice: once for $i \in \mathcal{V}(\overline{\Omega}_h)$ when $j \in \mathcal{N}_i$ and once for $j \in \mathcal{V}(\overline{\Omega}_h)$ when $i \in \mathcal{N}_j$. Thus, instead of a double sum over all nodes and all neighbors, we may equivalently sum over all edges and add contributions from both nodes associated with edge $\vec{i}j$. That is

$$\sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij} = \sum_{\vec{i}j \in \mathcal{E}(\overline{\Omega})} [\mathbf{e}_{ij} + \mathbf{e}_{ji}], \tag{B.30}$$

where $\mathcal{E}(\overline{\Omega})$ is the set of all edges \vec{ij} in the triangulation \mathcal{T}_h of the computational domain Ω_h .

Thus, applying the linearized operators in (B.28) to the variable \mathbf{q}_h and rewriting each double sum in (B.28) making use of (B.30), yields, for the convective terms

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega})} \frac{1}{2} \left[\mathbf{z}_i^T \frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \mathbf{q}_i + \mathbf{z}_i^T \frac{\partial(\mathbf{f}_j \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_j} \mathbf{q}_j \right. \\ \left. + \mathbf{z}_j^T \frac{\partial(\mathbf{f}_j \cdot \mathbf{n}_{ji})}{\partial \mathbf{w}_j} \mathbf{q}_j + \mathbf{z}_j^T \frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ji})}{\partial \mathbf{w}_i} \mathbf{q}_i \right] \end{aligned} \quad (\text{B.31})$$

Let us note that the aim of these transformations is to obtain, in the end, expressions where the term under the sum over all nodes i can be factorized by \mathbf{q}_i . Therefore, in (B.31) the products vector-matrix-vector are, first, transposed (these are scalars !), then, factorized by either \mathbf{q}_i or \mathbf{q}_j , to obtain

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega})} \frac{1}{2} \left[\mathbf{q}_i^T \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_i - \mathbf{z}_j) \right. \\ \left. + \mathbf{q}_j^T \left[\frac{\partial(\mathbf{f}_j \cdot \mathbf{n}_{ji})}{\partial \mathbf{w}_j} \right]^T (\mathbf{z}_j - \mathbf{z}_i) \right] \end{aligned} \quad (\text{B.32})$$

where it was used that $\mathbf{n}_{ij} = -\mathbf{n}_{ji}$ (A.4). Observe that the summation over the edges in (B.32) is of the same type than the summation on the right side of the identity (B.30). Making use of (B.30) in (B.32), we obtain

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \mathbf{q}_i^T \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_i - \mathbf{z}_j), \quad (\text{B.33})$$

making use of the definition of an adjoint (B.2), it yields

$$\left\langle \left(\frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h, \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \mathbf{q}_i^T \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_i - \mathbf{z}_j), \quad (\text{B.34})$$

which identifies the action of the adjoint of the linearized residuals of convection, at each node i , on the variable \mathbf{z}_h .

Proceeding with the terms involving boundary conditions,

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{z}_i^T \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right] \mathbf{q}_i, \quad (\text{B.35})$$

where the factorization by \mathbf{q}_i is obtained by transposing the product vector-matrix-vector ,

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{q}_i^T \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right]^T \mathbf{z}_i, \quad (\text{B.36})$$

which, by using (B.2), yields

$$\left\langle \left(\frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h, \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{q}_i^T \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right]^T \mathbf{z}_i, \quad (\text{B.37})$$

which shows how the adjoint of the residuals, of the fluxes imposing boundary conditions, apply to the variable \mathbf{z}_h .

The term involving the second order artificial dissipation are treated in the same fashion to obtain

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega})} \left[\mathbf{z}_i^T \sum_{l \in \mathcal{N}_i} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} \mathbf{q}_l + \mathbf{z}_i^T \sum_{k \in \mathcal{N}_j} \frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_k} \mathbf{q}_k \right. \\ \left. + \mathbf{z}_j^T \sum_{\alpha \in \mathcal{N}_j} \frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_\alpha} \mathbf{q}_\alpha + \mathbf{z}_j^T \sum_{\beta \in \mathcal{N}_i} \frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_\beta} \mathbf{q}_\beta \right], \end{aligned} \quad (\text{B.38})$$

At this stage, the development of the product involving the fourth order artificial viscosity is the same as the one for the second order artificial viscosity. Therefore, the results that will be obtained for the adjoint of second order artificial viscosity fluxes will be used to formulate the adjoint of the fourth order artificial viscosity. The products vector-matrix-vector in (B.38), are also transposed, which yields

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega})} \left[\sum_{l \in \mathcal{N}_i} \mathbf{q}_l^T \left[\frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} \right]^T \mathbf{z}_i + \sum_{k \in \mathcal{N}_j} \mathbf{q}_k^T \left[\frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_k} \right]^T \mathbf{z}_i \right. \\ \left. + \sum_{\alpha \in \mathcal{N}_j} \mathbf{q}_\alpha^T \left[\frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_\alpha} \right]^T \mathbf{z}_j + \sum_{\beta \in \mathcal{N}_i} \mathbf{q}_\beta^T \left[\frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_\beta} \right]^T \mathbf{z}_j \right]. \end{aligned} \quad (\text{B.39})$$

Sums over the neighboring nodes \mathcal{N}_i , and \mathcal{N}_j , respectively, are grouped under the same summation sign, respectively:

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega})} \left[\sum_{l \in \mathcal{N}_i} \mathbf{q}_l^T \left(\left[\frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} \right]^T \mathbf{z}_i + \left[\frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_l} \right]^T \mathbf{z}_j \right) \right. \\ \left. + \sum_{k \in \mathcal{N}_j} \mathbf{q}_k^T \left(\left[\frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_k} \right]^T \mathbf{z}_i + \left[\frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_k} \right]^T \mathbf{z}_j \right) \right]. \end{aligned} \quad (\text{B.40})$$

Making use of (B.10) we derive the following relations

$$\frac{\partial \mathbf{d}_{ij}^{\text{2nd}}}{\partial \mathbf{w}_l} = -\frac{\partial \mathbf{d}_{ji}^{\text{2nd}}}{\partial \mathbf{w}_l} \quad \forall l \in \mathcal{N}_i, \quad (\text{B.41})$$

and,

$$\frac{\partial \mathbf{d}_{ij}^{2\text{nd}}}{\partial \mathbf{w}_k} = -\frac{\partial \mathbf{d}_{ji}^{2\text{nd}}}{\partial \mathbf{w}_k} \quad \forall k \in \mathcal{N}_j. \quad (\text{B.42})$$

Note that similar relations hold for the fourth order artificial dissipation fluxes, that is, from (B.13) it holds that

$$\frac{\partial \mathbf{d}_{ij}^{4\text{th}}}{\partial \mathbf{w}_l} = -\frac{\partial \mathbf{d}_{ji}^{4\text{th}}}{\partial \mathbf{w}_l} \quad \forall l \in \mathcal{N}_i, \quad (\text{B.43})$$

and,

$$\frac{\partial \mathbf{d}_{ij}^{4\text{th}}}{\partial \mathbf{w}_k} = -\frac{\partial \mathbf{d}_{ji}^{4\text{th}}}{\partial \mathbf{w}_k} \quad \forall k \in \mathcal{N}_j. \quad (\text{B.44})$$

Relations (B.41)-(B.42) are used in (B.40) to obtain

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w &= \sum_{\vec{ij} \in \mathcal{E}(\bar{\Omega})} \left[\sum_{l \in \mathcal{N}_i} \mathbf{q}_l^T \left[\frac{\partial \mathbf{d}_{ij}^{2\text{nd}}}{\partial \mathbf{w}_l} \right]^T (\mathbf{z}_i - \mathbf{z}_j) \right. \\ &\quad \left. + \sum_{k \in \mathcal{N}_j} \mathbf{q}_k^T \left[\frac{\partial \mathbf{d}_{ji}^{2\text{nd}}}{\partial \mathbf{w}_k} \right]^T (\mathbf{z}_j - \mathbf{z}_i) \right]. \end{aligned} \quad (\text{B.45})$$

The sum over the edges, in expression (B.45), is transformed to a sum over nodes and neighboring nodes according to the identity (B.30),

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \sum_{j \in \mathcal{N}_i} \sum_{l \in \mathcal{N}_i} \mathbf{q}_l^T \left[\frac{\partial \mathbf{d}_{ij}^{2\text{nd}}}{\partial \mathbf{w}_l} \right]^T (\mathbf{z}_i - \mathbf{z}_j) \quad (\text{B.46})$$

Switching the order of the summations $j \in \mathcal{N}_i$ and $l \in \mathcal{N}_i$, in (B.46), and, switching the order of the summation $j \in \mathcal{N}_i$ and the variation \mathbf{q}_l , expression (B.46) is equivalent to

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \sum_{l \in \mathcal{N}_i} \mathbf{q}_l^T \sum_{j \in \mathcal{N}_i} \left[\frac{\partial \mathbf{d}_{ij}^{2\text{nd}}}{\partial \mathbf{w}_l} \right]^T (\mathbf{z}_i - \mathbf{z}_j) \quad (\text{B.47})$$

We repeat, in (B.47), the transformation of the double sum over the nodes to a sum over the edges (B.30), which yields

$$\begin{aligned} \left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w &= \sum_{il \in \mathcal{E}(\bar{\Omega})} \left[\mathbf{q}_l^T \sum_{j \in \mathcal{N}_i} \left[\frac{\partial \mathbf{d}_{ij}^{2\text{nd}}}{\partial \mathbf{w}_l} \right]^T (\mathbf{z}_i - \mathbf{z}_j) \right. \\ &\quad \left. + \mathbf{q}_i^T \sum_{k \in \mathcal{N}_l} \left[\frac{\partial \mathbf{d}_{lk}^{2\text{nd}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_l - \mathbf{z}_k) \right], \end{aligned} \quad (\text{B.48})$$

and using the identity (B.30) to transform (B.48) it is found that the expression (B.47) is equivalent to

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \sum_{l \in \mathcal{N}_i} \mathbf{q}_i^T \sum_{k \in \mathcal{N}_l} \left[\frac{\partial \mathbf{d}_{lk}^{2\text{nd}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_l - \mathbf{z}_k), \quad (\text{B.49})$$

that is,

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \mathbf{q}_i^T \sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_l} \left[\frac{\partial \mathbf{d}_{lk}^{2\text{nd}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_l - \mathbf{z}_k), \quad (\text{B.50})$$

where the action of the adjoint, of the second order artificial viscosity fluxes, on the variable \mathbf{z}_h is identified at each node i as we have

$$\left\langle \left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h, \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \mathbf{q}_i^T \sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_l} \left[\frac{\partial \mathbf{d}_{lk}^{2\text{nd}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_l - \mathbf{z}_k), \quad (\text{B.51})$$

Having the relations (B.43)-(B.44), we can use (B.50) to express $\langle \mathbf{z}_h, \partial \mathcal{D}_4 / \partial \mathbf{w}_h (\mathbf{q}_h) \rangle_w$, in (B.28), as

$$\left\langle \mathbf{z}_h, \frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \mathbf{q}_i^T \sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_l} \left[\frac{\partial \mathbf{d}_{lk}^{4\text{th}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_l - \mathbf{z}_k), \quad (\text{B.52})$$

which gives expression of the adjoint of the fourth order artificial viscosity

$$\left\langle \left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h, \mathbf{q}_h \right\rangle_w = \sum_{i \in \mathcal{V}(\bar{\Omega}_h)} \mathbf{q}_i^T \sum_{l \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_l} \left[\frac{\partial \mathbf{d}_{lk}^{4\text{th}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_l - \mathbf{z}_k), \quad (\text{B.53})$$

Expressions (B.34), (B.37), (B.50) and (B.53), define the adjoint of $\partial \mathcal{A}_h / \partial \mathbf{w}_h$, that is, for any \mathbf{z}_h

$$\left(\frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \equiv \left[\left(\frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \right)^* + \left(\frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \right)^* + \left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \right)^* + \left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \right)^* \right] \mathbf{z}_h, \quad (\text{B.54})$$

where the action of the adjoint operator on the variable \mathbf{z}_h is now detailed

$$\begin{aligned}
\left(\left(\frac{\partial \mathcal{P}}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \frac{1}{2} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_i - \mathbf{z}_j) & \forall i \in \mathcal{V}(\overline{\Omega}_h), \\
\left(\left(\frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i &= \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right]^T \mathbf{z}_i & \forall i \in \partial \mathcal{V}(\Omega_h), \\
\left(\left(\frac{\partial \mathcal{B}}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i &= \mathbf{0} & \forall i \in \mathcal{V}(\Omega_h), \\
\left(\left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_j} \left[\frac{\partial \mathbf{d}_{kj}^{\text{2nd}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_k - \mathbf{z}_j) & \forall i \in \mathcal{V}(\overline{\Omega}_h), \\
\left(\left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i &= \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_j} \left[\frac{\partial \mathbf{d}_{kj}^{\text{4th}}}{\partial \mathbf{w}_i} \right]^T (\mathbf{z}_k - \mathbf{z}_j) & \forall i \in \mathcal{V}(\overline{\Omega}_h).
\end{aligned} \tag{B.55}$$

The expressions of the adjoint of the linearized residuals due to the artificial dissipation, in (B.55), need to be developed. In the case of the fourth order artificial dissipation, we chose to use constant viscosities $\epsilon_{ij}^{\text{4th}}$. In the case of the second order artificial dissipation, the differentiation of the artificial viscosity is performed in §B.3.2. An alternative is to freeze the artificial viscosities of the second order fluxes, in which case the adjoint of the artificial dissipation is the same as we use self-adjoint operators for the diffusion.

The adjoint residuals for the artificial dissipation is now developed in the case of constant artificial viscosities. Making use of (B.10), for $\partial \epsilon_2^{ij} / \partial \mathbf{w}_l = 0$, for any index l , it yields

$$\left(\left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i = \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_j} \delta_{ik} \epsilon_2^{kj} (\mathbf{z}_k - \mathbf{z}_j) \quad \forall i \in \mathcal{V}(\overline{\Omega}_h), \tag{B.56}$$

where all terms under summation are zero if $k \neq i$, therefore it is rewritten as

$$\left(\left(\frac{\partial \mathcal{D}_2}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i = \sum_{j \in \mathcal{N}_i} \epsilon_2^{ij} (\mathbf{z}_i - \mathbf{z}_j) \quad \forall i \in \mathcal{V}(\overline{\Omega}_h). \tag{B.57}$$

Making use of (B.13), in expression (B.55), we can express the adjoint residuals issued of the fourth order artificial dissipation as

$$\left(\left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i = \sum_{j \in \mathcal{N}_i} \sum_{k \in \mathcal{N}_j} \left[-\epsilon_4^k \delta_{ki} \sum_{l \in \mathcal{N}_k} \mathbf{I} - \epsilon_4^j \mathbf{I} \right]^T (\mathbf{z}_k - \mathbf{z}_j) \quad \forall i \in \mathcal{V}(\overline{\Omega}_h), \quad (\text{B.58})$$

where terms that are indexed by k are zero unless $k = i$, thus simplifying expression (B.58) which is rewritten as

$$\left(\left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i = - \sum_{j \in \mathcal{N}_i} \epsilon_4^j \sum_{k \in \mathcal{N}_j} (\mathbf{z}_k - \mathbf{z}_j) - \sum_{j \in \mathcal{N}_i} \epsilon_4^i \sum_{l \in \mathcal{N}_i} (\mathbf{z}_i - \mathbf{z}_j) \quad \forall i \in \mathcal{V}(\overline{\Omega}_h). \quad (\text{B.59})$$

Switching the order of the summations $l \in \mathcal{N}_i$ and $j \in \mathcal{N}_i$, in (B.59), yields

$$\left(\left(\frac{\partial \mathcal{D}_4}{\partial \mathbf{w}_h} \right)^* \mathbf{z}_h \right)_i = - \sum_{j \in \mathcal{N}_i} \epsilon_4^j \nabla^2 \mathbf{z}_j + \sum_{l \in \mathcal{N}_i} \epsilon_4^i \nabla^2 \mathbf{z}_i \quad \forall i \in \mathcal{V}(\overline{\Omega}_h), \quad (\text{B.60})$$

where $\nabla^2 \mathbf{z}_i$ has previously been defined (3.23).

B.2.2 Adjoint equation and reduced gradient ∇f_n

So far, \mathbf{z}_h is an arbitrary variable in V_w . In the following, the example given in introduction of this chapter, expressed by the relations (4.1)-(4.6), is used as a guideline.

Let us recall that the discretization of the fluid domain Ω_h is described by the dual grid parameters (surface vectors) \mathbf{n}_h . The aim of the following demonstration is to express the variation of a function $f(\mathbf{w}_h, \mathbf{n}_h)$, subject to the discrete state equation 2.3, in terms of the variation of the grid parameters $\delta \mathbf{n}_h$.

We denote by \mathbf{w}_h^* the adjoint state, that is the solution of the *adjoint equation*

$$\left(\frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \right)^* \mathbf{w}_h^* = \frac{\partial f}{\partial \mathbf{w}_h}, \quad (\text{B.61})$$

where $(\partial \mathcal{A}_h / \partial \mathbf{w}_h)^*$ is defined by (B.55).

Given an *arbitrary* variation $\delta \mathbf{n}_h$, and defining $\delta \mathbf{w}_h$ as the solution of sensitivity equations (B.24), the variation of f is expressed as

$$\delta f = \left\langle \frac{\partial f}{\partial \mathbf{w}_h}, \delta \mathbf{w}_h \right\rangle_w + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n. \quad (\text{B.62})$$

Making use of the adjoint equation (B.61), (B.62) is expressed as

$$\delta f = \left\langle \left(\frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \right)^* \mathbf{w}_h^*, \delta \mathbf{w}_h \right\rangle_w + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n, \quad (\text{B.63})$$

which, by definition of an adjoint (B.2), is equivalent to

$$\delta f = \left\langle \mathbf{w}_h^*, \frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n. \quad (\text{B.64})$$

The sensitivity equations (B.24) hold, so that the equation (B.27) holds for an arbitrary \mathbf{z}_h , therefore, it holds for $\mathbf{z}_h = \mathbf{w}_h^*$, which yields

$$\left\langle \mathbf{w}_h^*, \frac{\partial \mathcal{A}_h}{\partial \mathbf{w}_h} \delta \mathbf{w}_h \right\rangle_w = - \left\langle \mathbf{w}_h^*, \frac{\partial \mathcal{A}_h}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w. \quad (\text{B.65})$$

Making use of (B.65), expression (B.64) is equivalent to

$$\delta f = - \left\langle \mathbf{w}_h^*, \frac{\partial \mathcal{A}_h}{\partial \mathbf{n}_h} \delta \mathbf{n}_h \right\rangle_w + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n. \quad (\text{B.66})$$

We replace $\partial \mathcal{A}_h / \partial \mathbf{n}_h$ above by its detailed expression (B.26) and use the definition of the inner product in V_w (B.1) in order to express (B.66) as

$$\begin{aligned} \delta f = & - \sum_{i \in \mathcal{V}(\overline{\Omega}_h)} \mathbf{w}_i^{*T} \sum_{j \in \mathcal{N}_i} \left[\mathbf{f}_{ij} \cdot \delta \mathbf{n}_{ij} + \left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n (\mathbf{w}_i - \mathbf{w}_j) \right] \\ & - \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{w}_i^{*T} \mathbf{f}_i^{\text{bc}} \cdot \delta \mathbf{n}_i + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n. \end{aligned} \quad (\text{B.67})$$

In order to identify the gradient ∇f_n , the double sum in (B.67) is changed to a single sum over all edges using identity (B.30), which yields

$$\begin{aligned} \delta f = & - \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega})} \left[\mathbf{w}_i^{*T} \mathbf{f}_{ij} \cdot \delta \mathbf{n}_{ij} + \mathbf{w}_j^{*T} \mathbf{f}_{ji} \cdot \delta \mathbf{n}_{ji} \right] \\ & + \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega})} \left[\left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n \mathbf{w}_i^{*T} (\mathbf{w}_i - \mathbf{w}_j) + \left\langle \frac{\partial \epsilon_2^{ji}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n \mathbf{w}_j^{*T} (\mathbf{w}_j - \mathbf{w}_i) \right] \\ & - \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{w}_i^{*T} \mathbf{f}_i^{\text{bc}} \cdot \delta \mathbf{n}_i + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n. \end{aligned} \quad (\text{B.68})$$

Using that $\mathbf{n}_{ij} = -\mathbf{n}_{ji}$ (A.4), $\epsilon_2^{ij} = \epsilon_2^{ji}$ (3.31), and $\mathbf{f}_{ij} = \mathbf{f}_{ji}$ (3.16), we rewrite (B.68) as

$$\begin{aligned} \delta f = & - \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega})} (\mathbf{w}_i^* - \mathbf{w}_j^*)^T \mathbf{f}_{ij} \cdot \delta \mathbf{n}_{ij} \\ & + \sum_{\vec{ij} \in \mathcal{E}(\overline{\Omega})} \left\langle \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n (\mathbf{w}_i^* - \mathbf{w}_j^*)^T (\mathbf{w}_i - \mathbf{w}_j) \\ & - \sum_{i \in \mathcal{V}(\partial \Omega_h)} \mathbf{w}_i^{*T} \mathbf{f}_i^{\text{bc}} \cdot \delta \mathbf{n}_i + \left\langle \frac{\partial f}{\partial \mathbf{n}_h}, \delta \mathbf{n}_h \right\rangle_n. \end{aligned} \quad (\text{B.69})$$

Using the sensitivities of the second order artificial viscosities (B.16)-(B.18) and, of the function f , with respect to variations in the surface vectors \mathbf{n}_h , yields expression of the gradient ∇f_n . In §B.3 the discrete sensitivities of the second order artificial viscosities are expressed 'almost everywhere'.

However, freezing these coefficients yields a simpler expression, which is an approximation of the exact gradient shown in §B.3. It is obtained from (B.69) by discarding the contribution of the dissipation terms.

B.3 Detailed expressions

B.3.1 Adjoint boundary conditions at farfield

Farfield boundary conditions are imposed in a weak form. A one-dimensional analysis of the characteristics is performed to calculate the flow state at the boundary \mathbf{v}^c and this state is used to compute the flux \mathbf{f}^{bc} at the boundary (3.18). The flow state \mathbf{v}^c is not differentiable everywhere because the sign function is used. The sign function is differentiable everywhere but at zero, so that, unless when an eigen-value is zero, the derivative of the sign function is simply zero. As a consequence, the differentiation (B.8) is exact (almost everywhere) unless, when, at a node, one of the characteristics is zero.

B.3.2 Adjoint artificial viscosity

The calculation of the fluxes (B.10) requires the differentiation of ϵ_2^{ij} (3.25), in its domain of differentiability.

It will be convenient to carry out the differentiation of (3.25)-(3.30) in primitive variables. The transformation to conservative variables is given by the relation

$$\frac{\partial \epsilon_2^{ij}}{\partial \mathbf{w}_k} = \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{v}_k} \frac{d\mathbf{v}_k}{d\mathbf{w}_k} \quad \forall k \in \mathcal{N}_i \text{ or } \forall k \in \mathcal{N}_j. \quad (\text{B.70})$$

The differentiation of the max and absolute function is defined almost everywhere. Let \mathbf{e} , \mathbf{f} and \mathbf{g} be three vectors in \mathbb{R}^d we will use

$$\frac{\partial |(e + f) \cdot g|}{\partial e} = \text{sign}((e + f) \cdot g) g \quad \text{if } (e + f) \cdot g \neq 0, \quad (\text{B.71})$$

and, for functions f and g of the variable x , we will use

$$\frac{d(\max(f, g))}{dx} = \begin{cases} \frac{df}{dx} & \text{if } f > g \\ \frac{dg}{dx} & \text{if } f < g \end{cases} \quad (\text{B.72})$$

In the following we consider $k \in \mathcal{N}_i$. Formulas for $k \in \mathcal{N}_j$ would be obtained by exchanging i by j , and j by i , in the expressions obtained for $k \in \mathcal{N}_i$. We apply the chain rule to the differentiation (3.25), in primitive variables, to obtain

$$\begin{aligned} \frac{\partial \epsilon_2^{ij}}{\partial \mathbf{v}_k} &= \epsilon_2 \frac{\partial \max(\text{psw}_i, \text{psw}_j)}{\partial \mathbf{v}_k} N_{ij} F_{ij} \\ &\quad + \epsilon_2 \max(\text{psw}_i, \text{psw}_j) N_{ij} \frac{\partial F_{ij}}{\partial \mathbf{v}_k} \end{aligned} \quad (\text{B.73})$$

From the definition of the pressure sensor (3.26) we obtain

$$\begin{aligned} \frac{\partial \text{psw}_i}{\partial \mathbf{v}_i} &= - \sum_{j \in \mathcal{N}_i} \frac{\partial p_i}{\partial \mathbf{v}_i} \frac{|\sum_{j \in \mathcal{N}_i} (p_i - p_j)|}{\left(\sum_{j \in \mathcal{N}_i} (p_i + p_j) + \epsilon\right)^2} \\ &\quad + \frac{1}{\sum_{j \in \mathcal{N}_i} (p_i + p_j) + \epsilon} \frac{\partial |\sum_{j \in \mathcal{N}_i} (p_i - p_j)|}{\partial \mathbf{v}_i} \\ \frac{\partial \text{psw}_i}{\partial \mathbf{v}_k} &= - \frac{\partial p_j}{\partial \mathbf{v}_k} \frac{|\sum_{j \in \mathcal{N}_i} (p_i - p_j)|}{\left(\sum_{j \in \mathcal{N}_i} (p_i + p_j) + \epsilon\right)^2} \\ &\quad + \frac{1}{\sum_{j \in \mathcal{N}_i} (p_i + p_j) + \epsilon} \frac{\partial |\sum_{j \in \mathcal{N}_i} (p_i - p_j)|}{\partial \mathbf{v}_k} \end{aligned} \quad (\text{B.74})$$

Using relations (3.27) and (3.28) yields

$$\begin{aligned} \frac{\partial F_{ij}}{\partial \mathbf{v}_k} &= 2 \frac{\beta_{ij} \beta_{ji}}{\beta_{ij} + \beta_{ji}} \left[\frac{\partial r_{ij}}{\partial \mathbf{v}_k} + \frac{\partial r_{ji}}{\partial \mathbf{v}_k} \right] \\ &\quad + 4 \frac{R_{ij}}{\beta_{ij} + \beta_{ji}} \left[\frac{\partial \beta_{ij}}{\partial \mathbf{v}_k} \beta_{ji} + \beta_{ij} \frac{\partial \beta_{ji}}{\partial \mathbf{v}_k} \right] \\ &\quad - 4 \beta_{ij} \beta_{ji} R_{ij} \frac{\frac{\partial \beta_{ij}}{\partial \mathbf{v}_k} + \frac{\partial \beta_{ji}}{\partial \mathbf{v}_k}}{(\beta_{ij} + \beta_{ji})^2}. \end{aligned} \quad (\text{B.75})$$

From the definition of r_{ij} (3.28) we have

$$\frac{\partial r_{ij}}{\partial \mathbf{v}_k} = 0, \quad (\text{B.76})$$

and

$$\frac{\partial r_{ji}}{\partial \mathbf{v}_k} = \frac{\partial |\mathbf{u}_j \cdot \mathbf{n}_{ij}|}{\partial \mathbf{v}_k} + \frac{\partial c_j}{\partial \mathbf{v}_k} |\mathbf{n}_{ij}| \neq 0 \quad \text{only if } k = j, \quad (\text{B.77})$$

where, for calorically perfect gas,

$$\frac{\partial c_k}{\partial \mathbf{v}_k} = \frac{dc_k}{d\mathbf{v}_k} = \begin{bmatrix} \frac{-c_k}{2\rho_k} \\ \mathbf{0} \\ \frac{c_k}{2p_k} \end{bmatrix}. \quad (\text{B.78})$$

The differentiation of β_{ij} (3.28) yields

$$\begin{aligned}\frac{\partial \beta_{ij}}{\partial \mathbf{v}_k} &= \frac{1}{4} \frac{\frac{\partial \lambda_i}{\partial \mathbf{v}_k}}{(\lambda_i R_{ij})^{1/2}} - \frac{1}{8} \frac{\frac{\partial r_{ji}}{\partial \mathbf{v}_k} (\lambda_i)^{1/2}}{(R_{ij})^{3/2}}, \\ \frac{\partial \beta_{ji}}{\partial \mathbf{v}_k} &= \frac{1}{4} \frac{\frac{\partial \lambda_j}{\partial \mathbf{v}_k}}{(\lambda_j R_{ji})^{1/2}} - \frac{1}{8} \frac{\frac{\partial r_{ji}}{\partial \mathbf{v}_k} (\lambda_j)^{1/2}}{(R_{ji})^{3/2}},\end{aligned}\tag{B.79}$$

Relation (3.29) yields

$$\begin{aligned}\frac{\partial \lambda_j}{\partial \mathbf{v}_k} &= \frac{1}{2} \sum_{l \in \mathcal{N}_j} \left[\frac{\partial |(\mathbf{u}_j + \mathbf{u}_l) \cdot \mathbf{n}_{jl}|}{\partial \mathbf{v}_k} + \frac{\partial c_j}{\partial \mathbf{v}_k} |\mathbf{n}_{jl}| \right] \quad \forall j \in \mathcal{V}(\Omega_h), \\ \frac{\partial \lambda_j}{\partial \mathbf{v}_k} &= \frac{1}{2} \sum_{l \in \mathcal{N}_j} \left[\frac{\partial |(\mathbf{u}_j + \mathbf{u}_l) \cdot \mathbf{n}_{jl}|}{\partial \mathbf{v}_k} + \frac{\partial c_j}{\partial \mathbf{v}_k} |\mathbf{n}_{jl}| \right] \\ &\quad + \frac{\partial |\mathbf{u}_j \cdot \mathbf{n}_j|}{\partial \mathbf{v}_k} + \frac{\partial c_j}{\partial \mathbf{v}_k} |\mathbf{n}_j| \quad \forall j \in \mathcal{V}(\partial \Omega_h),\end{aligned}\tag{B.80}$$

and

$$\frac{\partial \lambda_i}{\partial \mathbf{v}_k} = \frac{1}{2} \left[\frac{\partial |(\mathbf{u}_i + \mathbf{u}_k) \cdot \mathbf{n}_{ik}|}{\partial \mathbf{v}_k} + \frac{\partial c_k}{\partial \mathbf{v}_k} |\mathbf{n}_{ik}| \right] \quad \forall i \in \mathcal{V}(\overline{\Omega_h}).\tag{B.81}$$

B.3.3 Artificial viscosity in gradient expression

The differentiation of the second order artificial viscosity, with respect to the surface vectors, contributes to the gradient expression ∇f_n (B.69).

$$\frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_{il}} = \epsilon_2 \max(\text{psw}_i, \text{psw}_j) N_{ij} \frac{\partial F_{ij}}{\partial \mathbf{n}_{il}} \quad \forall l \in \mathcal{N}_i.\tag{B.82}$$

Using relations (3.27) and (3.28)

$$\begin{aligned}\frac{\partial F_{ij}}{\partial \mathbf{n}_{il}} &= 2 \frac{\beta_{ij} \beta_{ji}}{\beta_{ij} + \beta_{ji}} \left[\frac{\partial r_{ij}}{\partial \mathbf{n}_{il}} + \frac{\partial r_{ji}}{\partial \mathbf{n}_{il}} \right] \\ &\quad + 4 \frac{R_{ij}}{\beta_{ij} + \beta_{ji}} \left[\frac{\partial \beta_{ij}}{\partial \mathbf{n}_{il}} \beta_{ji} + \beta_{ij} \frac{\partial \beta_{ji}}{\partial \mathbf{n}_{il}} \right] \\ &\quad - 4 \beta_{ij} \beta_{ji} R_{ij} \frac{\frac{\partial \beta_{ij}}{\partial \mathbf{n}_{il}} + \frac{\partial \beta_{ji}}{\partial \mathbf{n}_{il}}}{(\beta_{ij} + \beta_{ji})^2}.\end{aligned}\tag{B.83}$$

From the definition of r_{ij} (3.28) we have

$$\frac{\partial r_{ij}}{\partial \mathbf{n}_{il}} = \frac{\partial |\mathbf{u}_i \cdot \mathbf{n}_{ij}|}{\partial \mathbf{n}_{il}} + c_i \frac{\partial |\mathbf{n}_{ij}|}{\partial \mathbf{n}_{il}},\tag{B.84}$$

which can be expressed using (B.71). Relation (3.29), differentiated with respect to the normal \mathbf{n}_{il} , gives

$$\frac{\partial \lambda_i}{\partial \mathbf{n}_{il}} = \frac{1}{2} \left[\frac{\partial |(\mathbf{u}_i + \mathbf{u}_l) \cdot \mathbf{n}_{il}|}{\partial \mathbf{n}_{il}} + (c_i + c_l) \frac{\partial |\mathbf{n}_{il}|}{\partial \mathbf{n}_{il}} \right] \quad \forall i \in \mathcal{V}(\overline{\Omega}_h). \quad (\text{B.85})$$

The differentiation with respect to \mathbf{n}_i , the normal vectors at the boundaries, is performed by applying the following relations

$$\frac{\partial \epsilon_2^{ij}}{\partial \mathbf{n}_i} = \epsilon_2 \max(\text{psw}_i, \text{psw}_j) N_{ij} \frac{\partial F_{ij}}{\partial \mathbf{n}_i}. \quad (\text{B.86})$$

Using relations (3.27) and (3.28)

$$\begin{aligned} \frac{\partial F_{ij}}{\partial \mathbf{n}_i} = & 4 \frac{R_{ij}}{\beta_{ij} + \beta_{ji}} \left[\frac{\partial \beta_{ij}}{\partial \mathbf{n}_i} \beta_{ji} + \beta_{ij} \frac{\partial \beta_{ji}}{\partial \mathbf{n}_i} \right] \\ & - 4 \beta_{ij} \beta_{ji} R_{ij} \frac{\frac{\partial \beta_{ij}}{\partial \mathbf{n}_i} + \frac{\partial \beta_{ji}}{\partial \mathbf{n}_i}}{(\beta_{ij} + \beta_{ji})^2}, \end{aligned} \quad (\text{B.87})$$

which requires the calculation of

$$\frac{\partial \lambda_i}{\partial \mathbf{n}_i} = \frac{\partial |\mathbf{u}_i \cdot \mathbf{n}_i|}{\partial \mathbf{n}_i} + c_i \frac{\partial |\mathbf{n}_i|}{\partial \mathbf{n}_i} \quad \forall i \in \mathcal{V}(\partial \Omega_h). \quad (\text{B.88})$$

Paper A

Shape optimization for delay of laminar-turbulent transition

Olivier Amoignon¹, Jan O. Pralits^{2,3}, Ardeshir Hanifi²,
Martin Berggren^{1,2} and Dan S. Henningson^{2,3}

¹Department of Information Technology, Uppsala University, Sweden

²Swedish Defense Research Agency, FOI, Aeronautics Division,
FFA, SE-172 90 Stockholm, Sweden

³Department of Mechanics, KTH, SE-100 44 Stockholm, Sweden

September, 2003

Abstract

Theory and results are presented for an approach to perform aerodynamic shape optimization with the aim of transition delay, and thus a decrease of the viscous drag. Linear stability theory is used to calculate the growth rate of perturbations with infinitely small amplitude, superimposed on the laminar mean flow. The location of laminar-turbulent transition is assumed to occur at the location where the total amplification of disturbances, with respect to the first streamwise position where the disturbance starts to grow, attains an empirically determined value, whose logarithm is generally denoted N . In the present approach, an iterative gradient-based optimization procedure is used with the aim of minimizing an objective function based on the disturbance kinetic energy. Here, the gradients of interest are efficiently evaluated using adjoint equations. The inviscid flow is obtained by solving the Euler equations for compressible flows, and the viscous mean flow is obtained from the solution of the boundary layer equations for compressible flows on infinite swept wings. The evolution of convectively unstable disturbances is analyzed using the linear parabolized stability equations (PSE). Large effort is used here to explain the state and adjoint equations involved, gradient evaluation, and validity tests of the gradient computed using the solution of the adjoint equations. Some first results are presented of shape optimization for transition delay. Numerical tests are carried out on the RAE 2822 airfoil and are formulated to reduce the disturbance kinetic energy while maintaining a fixed volume, angle of attack, leading edge radius and trailing edge position. Tests are also carried out to simultaneously reduce the wave drag and the disturbance kinetic energy while maintaining lift and pitch moment coefficients near their values at initial design.

1 Introduction

In the aeronautics industry, reducing the viscous drag on a wing while maintaining operational properties such as lift is of great interest, and the research in this area regarding active flow control is vast, see [24] for a thorough review on the topic of Laminar Flow Control (LFC). It is known that the viscous drag increases dramatically as the boundary layer flow changes from a laminar to a turbulent state. Therefore, an increase of the laminar portion of the wing, that is, moving downstream the point of laminar-turbulent transition, may decrease the viscous drag. Design of a geometry such that the laminar portion is increased or maximized is commonly denoted Natural Laminar Flow (NLF) design, which is a simpler and more robust technique than the ones devised in the area of active flow control. Once a feasible geometry is found, no additional devices such as suction systems, sensors, or actuators need to be mounted.

Transition in the boundary layer on aircraft wings is usually caused by breakdown of small disturbances that grow as they propagate downstream. The growth of these disturbances can be analyzed using linear stability theory, in which it is assumed that perturbations with infinitely small amplitude are superimposed on the laminar mean flow. The growth rate can then be used to predict the transition location using the so called e^N method, see [43, 40, 4]. In this method it is assumed that transition will occur at the location where the total amplification of the disturbance, with respect to the first streamwise position where the disturbance starts to grow, attains an empirically determined value, whose logarithm is generally denoted by N .

A distinctive feature of any flow design process is its computational cost. Despite that the complete flow field can in principle be obtained by solving the complete Navier-Stokes equations numerically, the computation is often very costly, or even totally out of reach for any existing computer when transitional and turbulent flow in complex geometries are involved. It is therefore common practice to introduce approximations. One approach, appropriate for flow over slender bodies, is to divide the flow into an inviscid outer flow field, and a viscous part describing the boundary layer at the surface. In this way, the growth rate of a disturbance superimposed on the boundary layer of a given geometry can be calculated as follows:

1. The solution of the equations describing the inviscid flow provides a pressure distribution on the surface of a given body.
2. The viscous mean flow is obtained by solving the boundary layer equations given the pressure distribution on the surface of the body and the geometry.
3. The linear stability equations are solved for a given mean flow and geometry, providing the growth rate.

Using this approach, three state equations must be solved (inviscid, boundary layer, linear stability), for any variation of the geometry in order to accurately predict the growth rate. However, if we consider the inviscid flow being the solution of the Euler equations, such a computation might be completed in minutes.

Independently from the strategy that enables to predict the growth rate of propagating disturbances for given geometry and flight data (Mach number, angle of attack, and Reynolds number), there are several approaches to the NLF design problem. Based on the knowledge how the growth rate of convectively unstable disturbances change due to variations of the pressure distribution, see for instance [38, 28, 46], an existing pressure distribution for a given wing can be altered manually followed by an analysis of the growth rate. Other approaches exist, where approximative relations have been derived between variations in the pressure distribution and N -factors [19]. Once a pressure distribution has been obtained that meets a given criteria regarding transition delay, a geometry must be designed which meets this “target” pressure distribution. Such an analysis is sometimes denoted *inverse design*, and was first studied by Lighthill [26] who solved it for the case of incompressible two-dimensional flow by conformal mapping. A review of different techniques to perform inverse design can be found in [22].

In this presentation, we investigate an optimal-control approach of a shape optimization problem that is formulated to perform NLF design. Optimal control theory concerns optimization problems ‘constrained’ by ordinary or partial differential equations (PDEs). The inverse design problem, mentioned above, may be attacked using an optimal-control approach: Given a flow model to compute the pressure distribution p on a surface Γ , that is, a system of PDEs with boundary conditions on Γ , find the shape $\hat{\Gamma}$ that minimizes a measure of the difference between the target pressure p^t and p . A relevant measure, called objective function in the context of optimization, is

$$J(p, \Gamma) = \frac{1}{2} \int_{\Gamma} |p(\mathbf{x}) - p^t(\mathbf{x})|^2 d\mathbf{x}. \quad (1)$$

The target pressure could, for example, be such that it damps the growth of disturbances as mentioned above. Let us denote

$$\mathcal{A}(p, \Gamma) = \mathbf{0} \quad (2)$$

the relation between the shape Γ , the system of PDEs with boundary conditions (on Γ), and its solution p . The optimization problem is summarized as

$$\min_{\Gamma \in \mathcal{F}} J(p, \Gamma) \quad \text{subject to } \mathcal{A}(p, \Gamma) = \mathbf{0}, \quad (3)$$

where \mathcal{F} is a set of admissible shapes. The system of PDEs, denoted \mathcal{A} , imposes constraints between the pressure distribution and the subject of

optimization, the shape Γ . The objective function J depends on the shape Γ as well as on the solution p of a PDE. Moreover, the pressure p is a function of Γ if it is the unique solution of the system (2). We denote by $\tilde{J}(\Gamma)$ the function defined by

$$\tilde{J}(\Gamma) = J(p(\Gamma), \Gamma), \quad (4)$$

where $p(\Gamma)$ is solution of the PDE (2). The problem (3) may then be reformulated in the *nested form*

$$\min_{\Gamma \in \mathcal{F}} \tilde{J}(\Gamma). \quad (5)$$

Gradient-based methods prove to be the most efficient for solving problems like (5), assumed that \tilde{J} (4) is continuous and differentiable. These methods are explained in details in text books on optimization, for example [32]. Meanwhile, the computation of the gradient of an objective function like \tilde{J} (4) is not trivial as J (1) depends explicitly on the design variable, here the shape Γ , as well as implicitly through the pressure, which is uniquely defined by the shape through the state equation (2). This is often a major difficulty in optimal-control problems. However, a practical method consists in approximating each component of the gradient $\nabla \tilde{J}$ by finite difference (each is a partial derivative). The cost is prohibitive for large scale problems, that is, for a large number of design parameters, due to expensive solutions of the PDEs (a 3D RANS solution may take days of computation). The cure is the solution of an additional PDE, called the adjoint problem, as it provides a means to calculate $\nabla \tilde{J}$ at a cost that is independent of the number of parameters. The use of adjoint equations in design optimization may be viewed as an off-spring of the theory of optimal control for PDE developed by J.L. Lions [27] in the 60's. Based on this approach, the optimal shape of a body in viscous flow at very low Reynolds number, called Stokes flows, could be derived by Pironneau [33] in 1973. In 1988, Jameson [22] formulated the adjoints of the full potential flow equations and of the Euler equations in order to solve inverse problems. Thereafter, research teams have developed adjoint codes for industrial applications to improve the design of aircrafts in which CFD codes are used for the flow computation (see [3, 6, 9, 14, 15, 23, 31, 16, 41, 42]). The reader will find an introduction to the method of adjoints, applied to aerodynamic design, in the reference [18] by Giles and Pierce. In our approach of NLF design, three systems of PDEs are solved sequentially, in order to calculate the objective function J , which is a function of the disturbance kinetic energy. The present work emphasizes the relation between the three adjoint problems that need to be solved for the calculation of the gradient ∇J .

Linear stability analysis has been used in the context of optimal NLF design in a number of investigations. In [19], an iterative approach uses a target pressure-N-factor relationship to compute the desired pressure distribution, and an inverse method to find the geometry which satisfies the

computed pressure distribution. The N -factor method has also been used in multidisciplinary optimization problems of whole aircraft configurations, where aerodynamics is considered as one discipline. In [25], it was used to predict the onset of transition in order to determine where to turn on a chosen turbulence model in the Reynolds-Averaged-Navier-Stokes equations, enabling calculation of the friction drag. In [30], a surface panel method was coupled with an approximative boundary layer calculation, and stability analysis. Note however, that none of these investigations explicitly calculated the sensitivities of a quantity obtained from the linear stability analysis such as the N -factor or disturbance kinetic energy, with respect to the geometry.

In the approach taken here, we use an iterative gradient-based optimization procedure [10] with the aim of minimizing an objective function, based on the disturbance kinetic energy, by changing the geometry of an airfoil. The inviscid flow is obtained by solving the Euler equations for compressible flows, and the viscous mean flow is obtained from the solution of the boundary layer equations for compressible flows on infinite swept wings. The evolution of convectively unstable disturbances is analyzed in the framework of nonlocal stability theory, which means that the growth of the boundary layer is taken into account, as opposed to the commonly used linear local stability theory. Indirectly, through minimization of the quadratic form associated with the Poisson equation, the design variables control the displacements of the nodes that are situated on the airfoil. Linear constraints on the displacements are accounted for in this parameterization so that they are fulfilled independently from the design variable. In this way we fix the volume, or cross section area, a limited region of the airfoil (around the leading edge) and the position of the trailing edge. The resulting quadratic programming formulation of the displacements is comparable to the formulation of the obstacle problems in mechanics. This technique generates smooth shapes at each optimization step without reducing the set of possible shapes, within the limit of the constraints and the size of the discretization. A simple mesh movement algorithm is used to propagate the nodal displacements from the airfoil to the rest of the computation domain of the inviscid flow. The gradient of the objective function is obtained from the solution of adjoint state equations, mesh sensitivities, and sensitivities of the parameterization.

The current report is an introduction and gives the status of an ongoing project on shape optimization for transition delay. Therefore a large effort is made to present the state and adjoint equations involved, gradient evaluation, and validity tests of the gradient computed using the solution of the adjoint equations. As this work is a joint project between one regarding shape optimization using the Euler equations, and another using the boundary layer and parabolized stability equations (PSE) for disturbance control, differences occur in the numerical schemes and methods used to derive the adjoint equations. Issues related to the latter is also discussed herein. Fi-

nally, some first results are presented of shape optimization for transition delay.

2 Theory

The aim of the current work is to perform gradient-based shape optimization in order to delay transition, and thus decrease the viscous drag, possibly in conjunction with wave-drag minimization. The objective function to be minimized is a measure of the disturbance growth. Its value is obtained by computing consecutively the inviscid flow for a given geometry, the viscous mean flow given the pressure distribution from the inviscid solution, and finally the linear growth rate for a given mean flow. In this section a concise description is given of the state equations involved, objective function based on the disturbance growth, gradient derivation, and resulting adjoint equations.

2.1 State equation for the inviscid flow

The system of Euler equations governs the flow of an inviscid compressible fluid and expresses the conservation of mass, momentum and energy. In steady state, the following integral form holds for any fixed region V with boundary ∂V

$$\int_{\partial V} \mathbf{f} \cdot \hat{\mathbf{n}} dS = 0, \quad (6)$$

where $\hat{\mathbf{n}}$ is the unit normal, outward oriented, of the control volume V , and \mathbf{f} is the 3-by-1 matrix of tensors

$$\mathbf{f} = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + I p \\ \mathbf{u} (E + p) \end{pmatrix}, \quad (7)$$

where E , the total energy per unit volume, is related to the pressure p , the density ρ and the velocity \mathbf{u} . In the framework of ideal fluids, assuming the law of perfect gas applies, E is given by

$$E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho \mathbf{u}^2. \quad (8)$$

In the case of an inviscid fluid, the condition

$$\mathbf{u} \cdot \hat{\mathbf{n}} = 0 \quad (9)$$

applies at the walls. The fluid state in conservative variables is denoted \mathbf{w} and is the 3-by-1 matrix

$$\mathbf{w} = \begin{pmatrix} \rho \\ \mathbf{m} \\ E \end{pmatrix}, \quad (10)$$

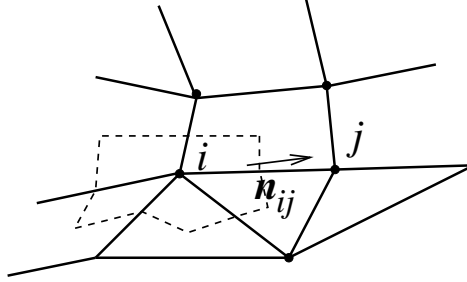


Figure 1: Dual grid (dashed lines) for inviscid flow computation.

where $\mathbf{m} = \rho \mathbf{u}$. Primitive variables are also used at some parts of the implementation and are denoted \mathbf{v} , the 3-by-1 matrix

$$\mathbf{v} = \begin{pmatrix} \rho \\ \mathbf{u} \\ p \end{pmatrix}. \quad (11)$$

In order to solve (6)-(9) for the flow around an airfoil, a finite sub-domain Ω is defined. Artificial boundary conditions are thus needed, and in the case of an airfoil, these are usually farfield conditions. We use the program Edge [13], a node-centered and edge-based finite-volume solver for Euler and the Reynolds Averaged Navier–Stokes equations (RANS). It is used here to solve equations (6)-(9) plus boundary conditions at the farfield.

Given a triangulation \mathcal{T}_h of the discrete domain Ω , we will denote by $\mathcal{V}(\overline{\Omega})$ the set of all node indexes, V_i the dual control volume at node i , and \mathbf{n}_{ij} the surface normal vector associated with edge ij (see Figure 1). For an introduction to this type of discretization, we refer to [5]. The set of edges is denoted $\mathcal{E}(\overline{\Omega})$. The steady state equations are solved by explicit time integration of the system

$$V_i \frac{d\mathbf{w}_i}{dt} + \mathbf{R}_i = 0, \quad \forall i \in \mathcal{V}(\overline{\Omega}), \quad (12)$$

until the residuals \mathbf{R}_i vanish within some tolerance. Convergence is accelerated by local time stepping, multigrid, and implicit residual smoothing. The residuals \mathbf{R}_i are

$$\begin{aligned} \mathbf{R}_i &= \sum_{j \in \mathcal{N}_i} (\mathbf{n}_{ij} \cdot \mathbf{f}_{ij} + \mathbf{d}_{ij}) & \forall i \in \mathcal{V}(\Omega), \\ \mathbf{R}_i &= \sum_{j \in \mathcal{N}_i} (\mathbf{n}_{ij} \cdot \mathbf{f}_{ij} + \mathbf{d}_{ij}) + \mathbf{n}_i \cdot \mathbf{f}_i^{\text{bc}} & \forall i \in \mathcal{V}(\partial\Omega). \end{aligned} \quad (13)$$

where \mathcal{N}_i is the set of indexes of nodes that are connected to node i with an edge. The residuals may be assembled by a single loop over all edges and

all boundary nodes [13]. The fluxes \mathbf{f} on a control surface associated with \mathbf{n}_{ij} is approximated by \mathbf{f}_{ij} which in this study is

$$\mathbf{f}_{ij} = \frac{1}{2}(\mathbf{f}_i + \mathbf{f}_j) \quad \text{with} \quad \mathbf{f}_i = \mathbf{f}(\mathbf{w}_i), \quad (14)$$

which gives a central scheme. An artificial dissipation flux \mathbf{d}_{ij} , a blend of second and fourth-order differences of Jameson type is used,

$$\mathbf{d}_{ij} = \epsilon_2^{ij}(\mathbf{w}_i - \mathbf{w}_j) + (\epsilon_4^i \nabla^2 \mathbf{w}_i - \epsilon_4^j \nabla^2 \mathbf{w}_j), \quad (15)$$

with

$$\nabla^2 \mathbf{w}_i = \sum_{k \in \mathcal{N}_i} (\mathbf{w}_k - \mathbf{w}_i) \quad \forall i \in \mathcal{V}(\bar{\Omega}). \quad (16)$$

The second-order dissipation is active where pressure gradients are large to prevent oscillations in the vicinity of shocks. The fourth-order dissipation is meant to remove oscillating solutions from grid point to grid point while preserving the second-order accuracy of the central scheme away from the shocks. The fluxes \mathbf{f} at the boundary are computed using a weak formulation of the boundary conditions [13] here denoted \mathbf{f}^{bc} . At node i on a wall, applying (9) for computing $\mathbf{f}_i \cdot \mathbf{n}_i$ amounts to taking

$$\mathbf{f}_i^{\text{bc}} = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} p_i \\ \mathbf{0} \end{pmatrix}. \quad (17)$$

At node i on a farfield boundary the fluxes (7) are computed using either the primitive farfield data \mathbf{v}_∞ , for incoming characteristics, or an extrapolation of the primitive variables \mathbf{v}_i for characteristics leaving the domain of computation:

$$\begin{aligned} \mathbf{f}_i^{\text{bc}} &= \mathbf{f}(\mathbf{v}_i^c(\hat{\mathbf{n}}_i)), \\ \mathbf{v}_i^c(\hat{\mathbf{n}}_i) &= \mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{H}(\lambda_i) \mathbf{L}^{-1}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{v}_i \\ &\quad + \mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) (\mathbf{I} - \mathbf{H}(\lambda_i)) \mathbf{L}^{-1}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{v}_\infty, \end{aligned} \quad (18)$$

where $\mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty)$ is a matrix of left eigenvectors that diagonalizes the Jacobian matrix of the flux in primitive variables along the outward-directed unit normal $\hat{\mathbf{n}}_i$, $\mathbf{H}(\lambda_i)$ is a diagonal matrix whose diagonal is 0 for negative eigenvalues and 1 for positive ones, and \mathbf{I} is the identity matrix.

2.2 State equations for the viscous flow

The flow field considered here is the boundary layer on a swept wing with infinite span, which is obtained by solving the mass, momentum, and energy conservation equations for a viscous compressible fluid. The equations are written in an orthogonal curvilinear coordinate system with streamwise,

spanwise, and wall-normal coordinates denoted as x^1, x^2 and x^3 respectively. A length element is defined as $ds^2 = (h_1 dx^1)^2 + (h_2 dx^2)^2 + (h_3 dx^3)^2$ where h_i is the scale factor. The total flow field, q_{tot} is decomposed into a mean, \bar{q} , and a perturbation part, \tilde{q} , as

$$q_{tot}(x^1, x^2, x^3, t) = \bar{q}(x^1, x^3) + \tilde{q}(x^1, x^2, x^3, t)$$

where $\bar{q} \in [U, V, W, P, T, \rho]$ and $\tilde{q} \in [\tilde{u}, \tilde{v}, \tilde{w}, \tilde{p}, \tilde{T}, \tilde{\rho}]$. Here U, V, W are the streamwise, spanwise and wall-normal velocity components of the mean flow, respectively, T is the temperature, ρ the density, and P the pressure. The lower case variables correspond to the disturbance quantities. The equations are derived for a quasi three-dimensional mean flow with zero variation in the spanwise direction. The evolution of convectively unstable disturbances is analyzed in the framework of the nonlocal stability theory. All flow and material quantities are made dimensionless with the corresponding reference flow quantities at a fixed streamwise position x_0^* , except the pressure, which is made dimensionless with twice the corresponding dynamic pressure. Here, dimensional quantities are indicated by the superscript \star . The reference length scale is taken as $l_0^* = (\nu_0^* x_0^* / U_0^*)^{\frac{1}{2}}$. The Reynolds and Mach number are defined as $Re = l_0^* U_0^* / \nu_0^*$ and $M = U_0^* / (\mathcal{R} \gamma T_0^*)^{\frac{1}{2}}$ respectively where \mathcal{R} is the specific gas constant, ν the kinematic viscosity and γ the ratio of the specific heats. In the proceeding sections the scale factors $h_2 = h_3 = 1$ are due to the infinite swept wing assumption.

2.2.1 Mean-flow equations

The dimensionless boundary-layer equations describing the steady viscous compressible mean flow on a swept wing with infinite span written on primitive variable form are given as

$$\frac{1}{h_1} \frac{\partial(\rho U)}{\partial x^1} + \frac{\partial(\rho W)}{\partial x^3} = 0, \quad (19)$$

$$\frac{\rho U}{h_1} \frac{\partial U}{\partial x^1} + \rho W \frac{\partial U}{\partial x^3} - \frac{1}{h_1} \frac{dP_e}{dx^1} + \frac{1}{Re} \frac{\partial}{\partial x^3} \left(\mu \frac{\partial U}{\partial x^3} \right) = 0, \quad (20)$$

$$\frac{\rho U}{h_1} \frac{\partial V}{\partial x^1} + \rho W \frac{\partial V}{\partial x^3} \frac{1}{Re} \frac{\partial}{\partial x^3} \left(\mu \frac{\partial V}{\partial x^3} \right) = 0, \quad (21)$$

$$\begin{aligned} & c_p \frac{\rho U}{h_1} \frac{\partial T}{\partial x^1} + c_p \rho W \frac{\partial T}{\partial x^3} + \frac{1}{Re Pr} \frac{\partial}{\partial x^3} \left(\kappa \frac{\partial T}{\partial x^3} \right) + \\ & (\gamma - 1) \frac{U M^2}{h_1} \frac{dP_e}{dx^1} + (\gamma - 1) \frac{\mu M^2}{Re} \left[\left(\frac{\partial U}{\partial x^3} \right)^2 + \left(\frac{\partial V}{\partial x^3} \right)^2 \right] = 0, \quad (22) \end{aligned}$$

where the dynamic viscosity is given by μ , specific heat at constant pressure c_p , and heat conductivity by κ . Under the boundary layer assumptions,

the pressure is constant in the direction normal to the boundary layer, i. e. $P = P_e(x^1)$. The equation of state can then be expressed as

$$\gamma M^2 P_e = \rho T,$$

and the streamwise derivative of the pressure is given as

$$\frac{dP_e}{dx^1} = -\rho_e U_e \frac{dU_e}{dx^1}$$

Here variables with subscript e are evaluated at the boundary layer edge. For a given pressure distribution given by the pressure coefficient

$$C_p = \frac{P^* - P_\infty^*}{\frac{1}{2}\rho_\infty^* Q_\infty^{*2}},$$

where Q_∞^* is the dimensional free stream velocity, and the sweep angle given by ψ , the values at the boundary layer edge are given as

$$P_e = \frac{P}{P_\infty} \frac{1}{\gamma M^2}, \quad T_e = \left(\frac{P}{P_\infty} \right)^{\frac{\gamma-1}{\gamma}}, \quad \rho_e = \left(\frac{P}{P_\infty} \right)^{\frac{1}{\gamma}}, \quad U_e = \sqrt{Q_e^2 - V_e^2},$$

where

$$\frac{P}{P_\infty} = 1 + \frac{1}{2} C_p \gamma M^2, \quad Q_e^2 = 1 + \frac{1 - T_e c_{p\infty}}{(\gamma - 1) \frac{1}{2} M^2}, \quad \text{and} \quad V_e = \sin \psi.$$

Here, we have used the assumptions that for an inviscid, steady, and adiabatic flow the total enthalpy is constant along a streamline, and the isentropic relations are used to obtain the relation between total and static quantities. A domain Ω_B is defined for equations (19)–(22) such that $x^1 \in [X_S, X_1]$, $x^2 \in [Z_0, Z_1]$ and $x^3 \in [0, \infty)$. The no-slip condition is used for the velocity components and the adiabatic wall condition for the temperature. In the free stream, the streamwise and spanwise velocity components, and the temperature takes the corresponding values at the boundary layer edge. This can be written as

$$\begin{aligned} \left[U, V, W, \frac{\partial T}{\partial x^3} \right] (x^1, 0) &= [0, 0, 0, 0] & \forall x^1 \in [X_S, X_1], \\ \lim_{x^3 \rightarrow +\infty} [U, V, T] (x^1, x^3) &= [U_e, V_e, T_e] (x^1) & \forall x^1 \in [X_S, X_1], \end{aligned} \quad (23)$$

These non-linear equations are solved in an iterative way. From (20)–(22) we obtain the solution of $\tilde{\mathbf{Q}} = (U, V, T)$ using the boundary condition above for a given value of W . Equation (19) is then integrated in the wall normal direction to obtain W . The solution is considered converged when the relative change in the wall-normal derivative of the streamwise velocity component at the wall is below a specified value. In the proceeding sections we consider $\mathbf{Q} = (U, V, W, T)$ to be the solution of the boundary layer state. This is made to simplify the presentation.

2.2.2 Disturbance equations

The perturbations are assumed to be time- and spanwise periodic waves as

$$\tilde{\mathbf{q}}(x^i, t) = \hat{\mathbf{q}}(x^1, x^3)\Theta, \quad \text{where} \quad \Theta = \exp\left(i \int_{X_0}^{x^1} \alpha(x') dx' + i\beta x^2 - i\omega t\right). \quad (24)$$

Here α is the complex streamwise wavenumber, β the real spanwise wavenumber, and ω the real disturbance angular frequency. Disturbances are superimposed on the mean flow at a streamwise position denoted X_0 . We assume a scale separation Re^{-1} between the weak variation in the x^1 -direction and the strong variation in the x^3 -direction. Further, it is assumed that $\partial/\partial x^1 \sim O(Re^{-1})$ and $W \sim O(Re^{-1})$. Introducing the ansatz given by equation (24) and the assumptions above in the linearized governing equations, keeping terms up to order $O(Re^{-1})$, yields a set of nearly parabolic partial differential equations [8, 29, 39, 21]. The system of equations, called Parabolized Stability Equations (PSE), are lengthy and therefore written here as

$$\mathcal{A}\hat{\mathbf{q}} + \mathcal{B}\frac{\partial\hat{\mathbf{q}}}{\partial x^3} + \mathcal{C}\frac{\partial^2\hat{\mathbf{q}}}{(\partial x^3)^2} + \mathcal{D}\frac{1}{h_1}\frac{\partial\hat{\mathbf{q}}}{\partial x^1} = 0, \quad (25)$$

where $\hat{\mathbf{q}} = (\hat{\rho}, \hat{u}, \hat{v}, \hat{w}, \hat{T})^T$. The coefficients of the 5×5 matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and \mathcal{D} are found in [35]. A domain Ω_p for equation (25) is defined such that $x^1 \in [X_0, X_1]$, $x^2 \in [Z_0, Z_1]$ and $x^3 \in [0, \infty)$. The boundary conditions corresponding to equation (25) are given as

$$\begin{aligned} \left[\hat{u}, \hat{v}, \hat{w}, \hat{T}\right](x^1, 0) &= [0, 0, 0, 0] \quad \forall x^1 \in [X_0, X_1], \\ \lim_{x^3 \rightarrow +\infty} \left[\hat{u}, \hat{v}, \hat{w}, \hat{T}\right](x^1, x^3) &= [0, 0, 0, 0] \quad \forall x^1 \in [X_0, X_1], \end{aligned}$$

To remove the ambiguity of having x^1 -dependence of both the amplitude and wave function in the ansatz, and to maintain a slow streamwise variation of the amplitude function $\hat{\mathbf{q}}$, a so called 'auxiliary condition' is introduced

$$\int_0^{+\infty} \hat{\mathbf{q}}^H \frac{\partial\hat{\mathbf{q}}}{\partial x^1} dx^3 = 0, \quad (26)$$

where superscript H denotes the complex conjugate transpose. Equation (25) is integrated in the downstream direction normal to the leading edge with an initial condition given by local stability theory. At each x^1 -position the streamwise wavenumber α is iterated such that the condition given by equation (26) is satisfied. After a converged streamwise wavenumber has been obtained, the growth rate of the disturbance kinetic energy can be calculated from the relation

$$\sigma = -\alpha_i + \frac{\partial}{\partial x^1}(\ln \sqrt{E}),$$

where

$$E = \int_0^{+\infty} \rho (|\hat{u}|^2 + |\hat{v}|^2 + |\hat{w}|^2) dx^3.$$

The growth rate can then be used to predict the transition location using the so called e^N -method [43, 40, 4]. The N -factor based on the disturbance kinetic energy is given as

$$N_E = \int_{X_{n1}}^X \sigma dx^1,$$

where X_{n1} is the lower branch of the neutral curve. A complete description of equation (25) is found in [35], and the numerical schemes used here are given in [20].

2.3 Objective function related to viscous drag

The objective here is to use shape optimization to reduce the viscous drag on a wing. A reduction of the viscous drag can be seen as an increase of the laminar portion of the wing, that is, to move the location of laminar-turbulent transition further downstream. It is therefore important that the chosen objective function can be related to the transition process. One choice is to measure the kinetic energy of a certain disturbance at a downstream position, say X_f . This can be written as

$$E_f = \frac{1}{2} \int_{Z_0}^{Z_1} \int_0^{+\infty} \tilde{\mathbf{q}}^H M \tilde{\mathbf{q}} h_1 dx^2 dx^3 \Big|_{x^1=X_f}, \quad (27)$$

where $M = \text{diag}(0, 1, 1, 1, 0)$ which means that the disturbance kinetic energy is calculated from the disturbance velocity components. If the position X_f is chosen as the upper branch of the neutral curve, then the measure can be related to the maximum value of the N -factor of a given disturbance as

$$N_{max} = \ln \sqrt{\frac{E_f}{E_0}}, \quad (28)$$

where E_0 is the disturbance kinetic energy at the first neutral point. If in addition, the value of the N -factor of the measured disturbance is the one which first reaches the transition N -factor, then the position can be related to the onset of laminar-turbulent transition. It is however not clear, a priori, that minimizing such a measure will damp the chosen or other disturbances in the whole unstable region, especially if different types of disturbances are present, such as TS and cross-flow waves. For Blasius flow, it has been shown that an objective function based on a single TS wave is sufficient to successfully damp the growth of other TS waves [37, 1]. On a wing however,

it is common that both TS and cross-flow waves are present simultaneously. An alternative is therefore to measure the kinetic energy as the streamwise integral over a defined domain. Using such an approach, several different disturbances can be accounted for, with respective maximum growth rate at different positions. Here, the size of K disturbances superimposed on the mean flow at an upstream position X_0 , is measured by their total kinetic energy as

$$E_K = \frac{1}{2} \sum_{k=1}^K \int_{X_{ms}}^{X_{me}} \int_{Z_0}^{Z_1} \int_0^{+\infty} \tilde{\mathbf{q}}_k^H M \tilde{\mathbf{q}}_k h_1 dx^1 dx^2 dx^3. \quad (29)$$

Here X_{ms} and X_{me} are the first and last streamwise position between which the disturbance kinetic energy is integrated, and adds the possibility to evaluate E_K in a streamwise domain within $[X_0, X_1]$. For a measure of a single disturbance, expression (29) is denoted E_1 .

2.4 Derivation of the gradient

The objective function evaluated for a single disturbance $J \equiv E_1$, expression (29), depends explicitly on $\tilde{\mathbf{q}}$ and on the (Euler) mesh¹, here defined by the vector of nodal coordinates \mathbf{X} , that is

$$J \equiv J(\tilde{\mathbf{q}}, \mathbf{X}). \quad (30)$$

The aim of our investigation is to minimize J (30), where $\tilde{\mathbf{q}}$ is the solution of the PSE (25). The latter is here given as

$$\mathcal{A}_q(\tilde{\mathbf{q}}, \mathbf{Q}, \mathbf{X}) = \mathbf{0}, \quad (31)$$

where (31) is defined for given \mathbf{X} and \mathbf{Q} . The mean flow \mathbf{Q} is solution of the BLE (19)-(22), here denoted

$$\mathcal{A}_Q(\mathbf{Q}, \mathbf{w}, \mathbf{X}) = \mathbf{0}, \quad (32)$$

which is defined for a given \mathbf{X} and \mathbf{w} . Finally, the inviscid flow \mathbf{w} is solution of the Euler equations (6)-(9), denoted

$$\mathcal{A}_w(\mathbf{w}, \mathbf{X}) = \mathbf{0}. \quad (33)$$

In the presentation of the adjoint problems it will be convenient to introduce the functions J_X , J_Q and J_w , defined in Table 1. These are just objective function (30) in which various intermediate quantities are regarded as independent variables.

¹The nodes on the airfoil are common to the three discretized equations: Euler, BLE and PSE.

Objective function:	$V_q \times V_X$ $\{\tilde{\mathbf{q}}, \mathbf{X}\}$	\longrightarrow	\mathbb{R} $J(\tilde{\mathbf{q}}, \mathbf{X})$	
subject to (31):	$V_Q \times V_X$ $\{\mathbf{Q}, \mathbf{X}\}$	\longrightarrow	\mathbb{R} $J_Q(\mathbf{Q}, \mathbf{X})$	$\equiv J(\tilde{\mathbf{q}}(\mathbf{Q}, \mathbf{X}), \mathbf{X})$
subject to (31)-(32):	$V_w \times V_X$ $\{\mathbf{w}, \mathbf{X}\}$	\longrightarrow	\mathbb{R} $J_w(\mathbf{w}, \mathbf{X})$	$\equiv J_Q(\mathbf{Q}(\mathbf{w}, \mathbf{X}), \mathbf{X})$
subject to (31)-(33):	V_X \mathbf{X}	\longrightarrow	\mathbb{R} $J_X(\mathbf{X})$	$\equiv J_w(\mathbf{w}(\mathbf{X}), \mathbf{X})$

Table 1: Functionals defined by the objective function J and the state equations (31)-(33).

The mesh nodes \mathbf{X} are calculated from the displacements \mathbf{y} of the nodes on the airfoil, by a mesh movement algorithm. This can be written $\mathbf{X} \equiv \mathbf{X}(\mathbf{y})$, and is described in §3. The displacements are controlled by the parameters \mathbf{a} , that is $\mathbf{y} \equiv \mathbf{y}(\mathbf{a})$, see §3. Given a function J_X of the variable \mathbf{X} , for example defined as in Table 1, it will also be convenient to define J_y and J_a , as

$$\begin{aligned} J_y(\mathbf{y}) &= J_X(\mathbf{X}(\mathbf{y})) , \\ J_a(\mathbf{a}) &= J_y(\mathbf{y}(\mathbf{a})) . \end{aligned} \tag{34}$$

To summarize our approach, the aim is to minimize J , subject to (31)-(33) with respect to the design parameters \mathbf{a} , using a gradient-based method. This requires the computation of the gradient ∇J_a which is computed from ∇J_X in §3. The aim of this section is to show that ∇J_X can be computed at an efficient cost using an optimal control approach.

In the following it is assumed that $\tilde{\mathbf{q}} \in V_q$, $\mathbf{Q} \in V_Q$, $\mathbf{w} \in V_w$, and $\mathbf{X} \in V_X$, and that V_q , V_Q , V_w and V_X are vector spaces equipped with the inner products $\langle \cdot, \cdot \rangle_q$, $\langle \cdot, \cdot \rangle_Q$, $\langle \cdot, \cdot \rangle_w$ and $\langle \cdot, \cdot \rangle_X$, respectively. Furthermore, it is assumed that all mappings are differentiable and, for example, $\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}}$ denotes linearization with respect to variable $\tilde{\mathbf{q}}$ of the mapping \mathcal{A}_q , at the given state $\{\tilde{\mathbf{q}}, \mathbf{Q}, \mathbf{w}, \mathbf{X}\}$. The notations $(\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}})^{-1}$, and $(\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}})^*$, denote the inverse, and the adjoint of the linearized mapping $\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}}$, respectively. Finally, the notation $(\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}}) \delta \tilde{\mathbf{q}}$ denotes the application of $\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}}$ on $\delta \tilde{\mathbf{q}}$.

2.4.1 Sensitivity of the PSE

For arbitrary variations $\{\delta \mathbf{Q}, \delta \mathbf{X}\} \in V_Q \times V_X$, of $\{\mathbf{Q}, \mathbf{X}\}$ in the PSE (31), the first variation of the solution of the PSE is denoted $\delta \tilde{\mathbf{q}} \in V_q$, and is

defined by the *sensitivity equations*

$$\frac{\partial \mathcal{A}_q}{\partial \tilde{\mathbf{q}}} \delta \tilde{\mathbf{q}} = -\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \delta \mathbf{Q} - \frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \delta \mathbf{X}. \quad (35)$$

Furthermore, for any variations $\{\delta \tilde{\mathbf{q}}, \delta \mathbf{X}\}$ in $V_q \times V_X$ we define the first variation of the objective function J as

$$\delta J = \left\langle \frac{\partial J}{\partial \tilde{\mathbf{q}}}, \delta \tilde{\mathbf{q}} \right\rangle_q + \left\langle \frac{\partial J}{\partial \mathbf{X}}, \delta \mathbf{X} \right\rangle_X \quad (36)$$

In the remaining, $\delta \tilde{\mathbf{q}}$ is solution of the sensitivity equations (35), which yields a new expression for (36)

$$\delta J = \left\langle \frac{\partial J}{\partial \tilde{\mathbf{q}}}, \left(\frac{\partial \mathcal{A}_q}{\partial \tilde{\mathbf{q}}} \right)^{-1} \left(-\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \delta \mathbf{Q} - \frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \delta \mathbf{X} \right) \right\rangle_q + \left\langle \frac{\partial J}{\partial \mathbf{X}}, \delta \mathbf{X} \right\rangle_X, \quad (37)$$

and, for $\tilde{\mathbf{q}}$ solution of (31) and $\delta \tilde{\mathbf{q}}$ solution of (35), the definition of J_Q (Table 1) yields

$$\delta J_Q = \delta J. \quad (38)$$

The gradient of the functional J_Q is $\nabla J_Q = \{\partial J_Q / \partial \mathbf{Q}, \partial J_Q / \partial \mathbf{X}\}$ and is a vector of the product space $V_Q \times V_X$ such that for all $\{\delta \mathbf{Q}, \delta \mathbf{X}\}$ in $V_Q \times V_X$ we have

$$\delta J_Q = \left\langle \frac{\partial J_Q}{\partial \mathbf{Q}}, \delta \mathbf{Q} \right\rangle_Q + \left\langle \frac{\partial J_Q}{\partial \mathbf{X}}, \delta \mathbf{X} \right\rangle_X. \quad (39)$$

Using the definition of the adjoint of the operator $\partial \mathcal{A}_q / \partial \tilde{\mathbf{q}}$ in expression (37) and using (38), we obtain

$$\begin{aligned} \delta J_Q = & \left\langle \left(\left(\frac{\partial \mathcal{A}_q}{\partial \tilde{\mathbf{q}}} \right)^{-1} \right)^* \frac{\partial J}{\partial \tilde{\mathbf{q}}}, -\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \delta \mathbf{Q} - \frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \delta \mathbf{X} \right\rangle_q \\ & + \left\langle \frac{\partial J}{\partial \mathbf{X}}, \delta \mathbf{X} \right\rangle_X, \end{aligned} \quad (40)$$

which is in turn rewritten using the definition of the adjoint of $\partial \mathcal{A}_q / \partial \mathbf{Q}$, and $\partial \mathcal{A}_q / \partial \mathbf{X}$, respectively, as

$$\begin{aligned} \delta J_Q = & - \left\langle \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \right)^* \left(\left(\frac{\partial \mathcal{A}_q}{\partial \tilde{\mathbf{q}}} \right)^{-1} \right)^* \frac{\partial J}{\partial \tilde{\mathbf{q}}}, \delta \mathbf{Q} \right\rangle_Q \\ & - \left\langle \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \left(\left(\frac{\partial \mathcal{A}_q}{\partial \tilde{\mathbf{q}}} \right)^{-1} \right)^* \frac{\partial J}{\partial \tilde{\mathbf{q}}}, \delta \mathbf{X} \right\rangle_X + \left\langle \frac{\partial J}{\partial \mathbf{X}}, \delta \mathbf{X} \right\rangle_X. \end{aligned} \quad (41)$$

Therefore, by introducing the adjoint state \mathbf{q}^* , solution of the system

$$\left(\frac{\partial \mathcal{A}_q}{\partial \tilde{\mathbf{q}}} \right)^* \mathbf{q}^* = \frac{\partial J}{\partial \tilde{\mathbf{q}}}, \quad (42)$$

we obtain that

$$\frac{\partial J_Q}{\partial \mathbf{Q}} = - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \right)^* \mathbf{q}^* \quad \text{and} \quad \frac{\partial J_Q}{\partial \mathbf{X}} = \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^*. \quad (43)$$

The cost for obtaining gradient of J_Q is reduced to one solution of the system (42) and two matrix–vector products as shown in (43).

2.4.2 Sensitivity of the BLE

For arbitrary variations $\{\delta \mathbf{w}, \delta \mathbf{X}\} \in V_w \times V_X$, of $\{\mathbf{w}, \mathbf{X}\}$ in the BLE (32), the first variation of the solution of the BLE is denoted $\delta \mathbf{Q} \in V_Q$, and is defined by the sensitivity equations

$$\frac{\partial \mathcal{A}_Q}{\partial \mathbf{Q}} \delta \mathbf{Q} = - \frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \delta \mathbf{w} - \frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \delta \mathbf{X}. \quad (44)$$

Furthermore, from the definition (39) and the expression of the gradient (43), for arbitrary variations $\{\delta \mathbf{Q}, \delta \mathbf{X}\}$ in $V_Q \times V_X$, the variation δJ_Q is

$$\delta J_Q = \left\langle - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \right)^* \mathbf{q}^*, \delta \mathbf{Q} \right\rangle_Q + \left\langle \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^*, \delta \mathbf{X} \right\rangle_X. \quad (45)$$

In the following, $\delta \mathbf{Q}$ is solution of the sensitivity equation (44). The variation δJ_Q is expressed, making use of (45) and (44), as

$$\begin{aligned} \delta J_Q = & \left\langle - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \right)^* \mathbf{q}^*, \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{Q}} \right)^{-1} \left(- \frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \delta \mathbf{w} - \frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \delta \mathbf{X} \right) \right\rangle_Q \\ & + \left\langle \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^*, \delta \mathbf{X} \right\rangle_X, \end{aligned} \quad (46)$$

and, for \mathbf{Q} solution of (32) and $\delta \mathbf{Q}$ solution of (44), the definition of J_w (Table 1) yields

$$\delta J_w = \delta J_Q. \quad (47)$$

The gradient of J_w is the vector $\{\partial J_w / \partial \mathbf{w}, \partial J_w / \partial \mathbf{X}\}$ in the product space $V_w \times V_X$ such that for all $\{\delta \mathbf{w}, \delta \mathbf{X}\}$ in $V_w \times V_X$ we have

$$\delta J_w = \left\langle \frac{\partial J_w}{\partial \mathbf{w}}, \delta \mathbf{w} \right\rangle_w + \left\langle \frac{\partial J_w}{\partial \mathbf{X}}, \delta \mathbf{X} \right\rangle_X. \quad (48)$$

Using the adjoint of the inverse linearized BLE operator $(\partial \mathcal{A}_Q / \partial \mathbf{Q})^{-1}$ in (46), δJ_w (47) is expressed as

$$\begin{aligned} \delta J_w = & \left\langle - \left(\left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{Q}} \right)^{-1} \right)^* \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{Q}} \right)^* \mathbf{q}^*, - \frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \delta \mathbf{w} - \frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \delta \mathbf{X} \right\rangle_Q \\ & + \left\langle \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^*, \delta \mathbf{X} \right\rangle_X. \end{aligned} \quad (49)$$

Using the adjoints of $\partial\mathcal{A}_Q/\partial\mathbf{w}$ and $\partial\mathcal{A}_Q/\partial\mathbf{X}$ enables us to rewrite relation (49) as

$$\begin{aligned}\delta J_w = & \left\langle \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{w}} \right)^* \left(\left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{Q}} \right)^{-1} \right)^* \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{Q}} \right)^* \mathbf{q}^*, \delta\mathbf{w} \right\rangle_w \\ & + \left\langle \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{X}} \right)^* \left(\left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{Q}} \right)^{-1} \right)^* \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{Q}} \right)^* \mathbf{q}^*, \delta\mathbf{X} \right\rangle_X \\ & + \left\langle \frac{\partial J}{\partial\mathbf{X}} - \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{X}} \right)^* \mathbf{q}^*, \delta\mathbf{X} \right\rangle_X.\end{aligned}\quad (50)$$

It suggests, as previously, to define an adjoint state \mathbf{Q}^* as

$$\left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{Q}} \right)^* \mathbf{Q}^* = \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{Q}} \right)^* \mathbf{q}^*. \quad (51)$$

Setting \mathbf{Q}^* in (50) and identifying the new expression with (48) we obtain

$$\frac{\partial J_w}{\partial\mathbf{w}} = \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{w}} \right)^* \mathbf{Q}^* \quad \text{and} \quad \frac{\partial J_w}{\partial\mathbf{X}} = \frac{\partial J}{\partial\mathbf{X}} - \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{X}} \right)^* \mathbf{Q}^*. \quad (52)$$

The use of adjoint equations limits cost for obtaining the gradient of J_w to solving the systems (42) and (51), as well as four matrix–vector products: one to ‘assemble’ the right-hand side of the adjoint system (51) and three to obtain the final expression (52).

2.4.3 Sensitivity of the Euler equations

For arbitrary variations $\delta\mathbf{X} \in V_X$, of \mathbf{X} in the Euler equation (33), the first variation of solution of the Euler equation is denoted $\delta\mathbf{w} \in V_w$, and is defined by the sensitivity equation

$$\frac{\partial\mathcal{A}_w}{\partial\mathbf{w}}\delta\mathbf{w} = -\frac{\partial\mathcal{A}_w}{\partial\mathbf{X}}\delta\mathbf{X}. \quad (53)$$

Furthermore, for arbitrary variations $\{\delta\mathbf{w}, \delta\mathbf{X}\}$ in $V_w \times V_X$ the first variation of the functional J_w is expressed, from the gradient (52)

$$\delta J_w = \left\langle \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{w}} \right)^* \mathbf{Q}^*, \delta\mathbf{w} \right\rangle_w + \left\langle \frac{\partial J}{\partial\mathbf{X}} - \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{X}} \right)^* \mathbf{Q}^*, \delta\mathbf{X} \right\rangle_X \quad (54)$$

In the following, $\delta\mathbf{w}$ is the solution of the sensitivity equation (53), which enables us to rewrite expression (54) as

$$\begin{aligned}\delta J_w = & \left\langle \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{w}} \right)^* \mathbf{Q}^*, -\left(\frac{\partial\mathcal{A}_w}{\partial\mathbf{w}} \right)^{-1} \frac{\partial\mathcal{A}_w}{\partial\mathbf{X}} \delta\mathbf{X} \right\rangle_w \\ & + \left\langle \frac{\partial J}{\partial\mathbf{X}} - \left(\frac{\partial\mathcal{A}_q}{\partial\mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial\mathcal{A}_Q}{\partial\mathbf{X}} \right)^* \mathbf{Q}^*, \delta\mathbf{X} \right\rangle_X,\end{aligned}\quad (55)$$

and, for \mathbf{w} solution of (33) and $\delta\mathbf{w}$ solution of (53), the definition of J_X (Table 1) yields

$$\delta J_X = \delta J_w. \quad (56)$$

The gradient of J_X is the vector ∇J_X in the space V_X such that for all $\delta\mathbf{X}$ in V_X we have

$$\delta J_X = \langle \nabla J_X, \delta\mathbf{X} \rangle_X. \quad (57)$$

The adjoint of the linearized Euler operator is used in (55) to express δJ_X (56) as

$$\begin{aligned} \delta J_X = & \left\langle \left(\left(\frac{\partial \mathcal{A}_w}{\partial \mathbf{w}} \right)^{-1} \right)^* \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \right)^* \mathbf{Q}^*, -\frac{\partial \mathcal{A}_w}{\partial \mathbf{X}} \delta\mathbf{X} \right\rangle_w \\ & + \left\langle \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \right)^* \mathbf{Q}^*, \delta\mathbf{X} \right\rangle_X. \end{aligned} \quad (58)$$

The adjoint instead of the linear operator $\partial \mathcal{A}_w / \partial \mathbf{X}$ is used in (58) and leads to

$$\begin{aligned} \delta J_X = & \left\langle \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \right)^* \mathbf{Q}^*, \delta\mathbf{X} \right\rangle_X \\ & - \left\langle \left(\frac{\partial \mathcal{A}_w}{\partial \mathbf{X}} \right)^* \left(\left(\frac{\partial \mathcal{A}_w}{\partial \mathbf{w}} \right)^{-1} \right)^* \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \right)^* \mathbf{Q}^*, \delta\mathbf{X} \right\rangle_X, \end{aligned} \quad (59)$$

The method of adjoint is again applied as we define an adjoint state \mathbf{w}^* , here solution of the system

$$\left(\frac{\partial \mathcal{A}_w}{\partial \mathbf{w}} \right)^* \mathbf{w}^* = \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \right)^* \mathbf{Q}^*, \quad (60)$$

which enables us to give expression for the gradient (57)

$$\nabla J_X = \frac{\partial J}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \right)^* \mathbf{Q}^* - \left(\frac{\partial \mathcal{A}_w}{\partial \mathbf{X}} \right)^* \mathbf{w}^*. \quad (61)$$

The total cost of this gradient evaluation is three adjoint systems (42), (51) and (60), and five matrix–vector products: two for the assembly of the right-hand-sides of the systems (51) and (60), and three for the final expression (61).

2.5 Adjoint equations

The concise description given in §2.4 gives an expression of the gradient ∇J_X (61), which is a function of three adjoint states \mathbf{q}^* , \mathbf{Q}^* and \mathbf{w}^* . These states are the solutions of the adjoint of the parabolized stability equations (42), adjoint of the boundary layer equations (51) and adjoint of the Euler

equations (60), respectively. There are in principle two different approaches on how to derive these adjoint equations. In the first, sometimes denoted the “discrete approach” or “discretize-then-differentiate”, the adjoint equations are derived from the discretized set of state equations. In the second approach, sometimes denoted the “continuous approach” or “differentiate-then-discretize”, the adjoint equations are derived from the continuous state equations. The continuous adjoint equations are then discretized, commonly in a similar way as the corresponding state equations, in which the original code written for solving the state is reused with some modifications.

2.5.1 Adjoint of the Parabolized Stability Equations

The adjoint of the parabolized stability equations (42) are derived using a continuous approach. The complete derivation is found in [35] and they are here given as

$$\tilde{\mathcal{A}}\mathbf{q}^* + \tilde{\mathcal{B}}\frac{\partial\mathbf{q}^*}{\partial x^3} + \tilde{\mathcal{C}}\frac{\partial^2\mathbf{q}^*}{(\partial x^3)^2} + \tilde{\mathcal{D}}\frac{1}{h_1}\frac{\partial\mathbf{q}^*}{\partial x^1} = \mathbf{S}_P^*, \quad (62)$$

$$\begin{aligned} & \frac{\partial}{\partial x^1} \int_0^{+\infty} \mathbf{q}^{*H} \left(\frac{\partial\mathcal{A}}{\partial\alpha} + \frac{\partial\mathcal{B}}{\partial\alpha} \right) \hat{\mathbf{q}} h_1 dx^3 = \\ & \begin{cases} 0 & \forall x^1 \notin [X_{ms}, X_{me}], \\ -i|\Theta|^2 \int_0^{+\infty} \hat{\mathbf{q}}^H M \hat{\mathbf{q}} h_1 dx^3 & \forall x^1 \in [X_{ms}, X_{me}], \end{cases} \end{aligned} \quad (63)$$

where

$$\mathbf{S}_P^* = \begin{cases} -\tilde{r}^* \frac{\partial\hat{\mathbf{q}}}{\partial x^1} - \frac{\partial(r^*\hat{\mathbf{q}})}{\partial x^1} & \forall x^1 \notin [X_{ms}, X_{me}], \\ -\tilde{r}^* \frac{\partial\hat{\mathbf{q}}}{\partial x^1} - \frac{\partial(r^*\hat{\mathbf{q}})}{\partial x^1} + \xi M^H \hat{\mathbf{q}} |\Theta|^2 & \forall x^1 \in [X_{ms}, X_{me}], \end{cases}$$

and

$$\begin{aligned} \tilde{\mathcal{A}} &= \mathcal{A}^H - \frac{\partial\mathcal{B}^H}{\partial x^3} - m_{13} \mathcal{B}^H + \frac{\partial^2\mathcal{C}^H}{(\partial x^3)^2} + 2 m_{13} \frac{\partial\mathcal{C}^H}{\partial x^3} - \frac{\partial\mathcal{D}^H}{\partial x^1}, \\ \tilde{\mathcal{B}} &= -\mathcal{B}^H + 2 \frac{\partial\mathcal{C}^H}{\partial x^3} + 2 m_{13} \mathcal{C}^H, \\ \tilde{\mathcal{C}} &= \mathcal{C}^H, \\ \tilde{\mathcal{D}} &= -\mathcal{D}^H. \end{aligned}$$

Here, $\mathbf{q}^* = (\rho^*, u^*, v^*, w^*, \theta^*)^T$, and the above equations are subject to the following boundary conditions

$$\begin{aligned} [u^*, v^*, w^*, \theta^*](x^1, 0) &= [0, 0, 0, 0] & \forall x^1 \in [X_0, X_1], \\ \lim_{x^3 \rightarrow +\infty} [u^*, v^*, w^*, \theta^*](x^1, x^3) &= [0, 0, 0, 0] & \forall x^1 \in [X_0, X_1]. \end{aligned}$$

The initial conditions are

$$\begin{aligned} \mathbf{q}^*(X_1, x^3) &= (1 - \xi)\mathbf{q}_1^*(x^3) & \forall x^3 \in [0, +\infty), \\ r^*(X_1) &= (1 - \xi)r_1^* & \forall x^3 \in [0, +\infty), \end{aligned}$$

with \mathbf{q}_1^* and r_1^* evaluated at $x^1 = X_1$ as

$$\begin{aligned} \mathbf{q}_1^* &= |\Theta|^2 \mathcal{D}^+(M - c_1 \mathcal{I}) \hat{\mathbf{q}}, & r_1^* &= |\Theta|^2 c_1, \\ \bar{c}_1 &= \frac{\int_0^\infty (h_1 \hat{\mathbf{q}}^H M \mathcal{D}^{+H} \left(\frac{\partial \mathcal{A}}{\partial \alpha} + \frac{\partial \mathcal{B}}{\partial \alpha} \right) \hat{\mathbf{q}} - i \hat{\mathbf{q}}^H M \hat{\mathbf{q}}) dx^3}{\int_0^\infty \hat{\mathbf{q}}^H \mathcal{D}^{+H} \left(\frac{\partial \mathcal{A}}{\partial \alpha} + \frac{\partial \mathcal{B}}{\partial \alpha} \right) \hat{\mathbf{q}} h_1 dx^3}, \end{aligned} \quad (64)$$

where $\mathcal{D}^+ = (\mathcal{D}^H)^{-1}$. Equation (62) is solved by backward integration in space. Even though it is a linear equation, in order to reuse the code developed to solve the PSE iteratively to satisfy the auxiliary condition, at each streamwise position r^* is solved iteratively such that expression (63) is satisfied. The right hand side of equation (62) and the initial condition depend on the choice of objective function, i. e. the value of ξ .

2.5.2 Adjoint of the Boundary Layer Equations

The adjoint of the boundary-layer equations are derived using a continuous approach. Details regarding the derivation is found in [34, 36] and they are here given as

$$\rho \frac{\partial(h_1 W^*)}{\partial x^3} - h_1 \rho \left(\frac{\partial U}{\partial x^3} U^* + \frac{\partial V}{\partial x^3} V^* + c_p \frac{\partial T}{\partial x^3} T^* \right) = S_W^*, \quad (65)$$

$$\begin{aligned} & \frac{\partial(\rho U U^*)}{\partial x^1} + \frac{\partial(h_1 \rho W U^*)}{\partial x^3} - \rho \left(\frac{\partial U}{\partial x^1} U^* + \frac{\partial V}{\partial x^1} V^* - \frac{\partial W^*}{\partial x^1} + c_p \frac{\partial T}{\partial x^1} T^* \right) + \\ & (\gamma - 1) M^2 \frac{dP_e}{dx^1} T^* - \frac{2(\gamma - 1)}{Re} M^2 \frac{\partial}{\partial x^3} \left(h_1 \mu \frac{\partial U}{\partial x^3} T^* \right) + \\ & \frac{1}{Re} \frac{\partial}{\partial x^3} \left(\mu \frac{\partial(h_1 U^*)}{\partial x^3} \right) = S_U^*, \end{aligned} \quad (66)$$

$$\begin{aligned} & \frac{\partial(\rho U V^*)}{\partial x^1} + \frac{\partial(h_1 \rho W V^*)}{\partial x^3} - \frac{2(\gamma - 1)}{Re} M^2 \frac{\partial}{\partial x^3} \left(h_1 \mu \frac{\partial V}{\partial x^3} T^* \right) + \\ & \frac{1}{Re} \frac{\partial}{\partial x^3} \left(\mu \frac{\partial(h_1 V^*)}{\partial x^3} \right) = S_V^*, \end{aligned} \quad (67)$$

$$\begin{aligned} & c_p \frac{\partial(\rho U T^*)}{\partial x^1} + c_p \frac{\partial(h_1 \rho W T^*)}{\partial x^3} + \frac{\rho U}{T} \left(\frac{\partial U}{\partial x^1} U^* + \frac{\partial V}{\partial x^1} V^* - \frac{\partial W^*}{\partial x^1} \right) + \\ & \frac{\rho U}{T} c_p \frac{\partial T}{\partial x^1} T^* + \frac{\kappa}{Re Pr} \frac{\partial^2(h_1 T^*)}{(\partial x^3)^2} \frac{(\gamma - 1)}{Re} M^2 \frac{d\mu}{dT} \left[\left(\frac{\partial U}{\partial x^3} \right)^2 + \left(\frac{\partial V}{\partial x^3} \right)^2 \right] T^* - \\ & \frac{1}{Re} \frac{d\mu}{dT} \left[\frac{\partial U}{\partial x^3} \frac{\partial(h_1 U^*)}{\partial x^3} + \frac{\partial V}{\partial x^3} \frac{\partial(h_1 V^*)}{\partial x^3} \right] = S_E^*, \end{aligned} \quad (68)$$

where the right hand side $\mathbf{S}_B^* = (S_W^*, S_U^*, S_V^*, S_E^*)^T$ is given as

$$\mathbf{S}_B^* = \begin{cases} (F_W, F_U, F_V, F_T + F_W W/T)^T h_1 & \forall x^1 \in (X_0, X_1), \\ \mathbf{0} & \forall x^1 \in (X_S, X_0], \end{cases} \quad (69)$$

The non-zero right hand side is the coupling between the APSE and the ABLE and express the sensitivity of the PSE with respect the variations in W , U , V and T respectively. A detailed description is found in [34]. The above equations are subjected to the following boundary conditions

$$\begin{aligned} \left[U^*, V^*, \frac{\partial(h_1 T^*)}{\partial x^3} \right] (x^1, 0) &= [0, 0, 0] & \forall x^1 \in [X_0, X_1], \\ \lim_{x^3 \rightarrow +\infty} [U^*, V^*, W^*, T^*] (x^1, x^3) &= [0, 0, 0, 0] & \forall x^1 \in [X_0, X_1]. \end{aligned}$$

The initial condition at $x^1 = X_1$ is given as

$$\mathbf{Q}^*(X_1, x^3) = \mathbf{0}. \quad \forall x^3 \in [0, +\infty),$$

These equations are linear oppose to the BLE but are however solved in a similar iterative way as was outlined in §2.2.1 in order to reuse the existing solver for the boundary layer equations. The ABLE are solved by backward integration in the streamwise direction. At each streamwise position we obtain the solution of $\tilde{\mathbf{Q}}^* = (U^*, V^*, T^*)$ from equations (66)–(68) using the boundary conditions above for a given value of W^* . Equation (65) is integrated in the wall-normal direction in order to obtain W^* . The solution at each streamwise position is considered converged when the relative variation of W^* is below a specified value.

The coupling between the Euler and the boundary-layer equations is the pressure distribution P_e and the mesh given by the nodal coordinates \mathbf{X} . A variation of the geometry which affects the Euler solution, will therefore appear as variations of the pressure distribution in the boundary layer equations, which consequently will affect the solution of the stability equations. In [37], the possibility of an optimal control problem using the pressure distribution as control variables and the total disturbance kinetic energy as the objective function, was considered. From the coupled APSE and ABLE for incompressible flows, an expression was derived for the gradient of the objective function with respect to the pressure distribution. From the present APSE and ABLE a similar expression can be evaluated and is here given as

$$\nabla J_P = \int_0^{+\infty} \left(-\frac{\partial U^*}{\partial x^1} + (\gamma - 1) M^2 \frac{\partial(T^* U)}{\partial x^1} \right) dx^3 \quad \forall x^1 \in (X_S, X_1). \quad (70)$$

Setting the Mach number equal to zero in expression (70), we find exactly the same expression as the one derived in [37]. Note that a variation of \mathbf{X} will also affect the nodal coordinates of the BLE and PSE. This can be seen in expression (61).

2.5.3 Adjoint of the Euler equations

The adjoint equations (60) are solved following the same technique used for solving the Euler equations (§2.1), by explicit time integration of the system

$$V_i \frac{d\mathbf{w}_i^*}{dt} + \mathbf{R}_i^* = \mathbf{0}, \quad \forall i \in \mathcal{V}(\overline{\Omega}) \quad (71)$$

until the residuals \mathbf{R}_i^* vanish within some tolerance. Derivation of the adjoint of the Euler equations can be found in [2]. The following gives expression for the adjoint residuals

$$\begin{aligned} \mathbf{R}_i^* &= \sum_{j \in \mathcal{N}_i} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T \frac{(\mathbf{w}_i^* - \mathbf{w}_j^*)}{2} + \sum_{j \in \mathcal{N}_i} \mathbf{d}_{ij}^* \quad \forall i \in \mathcal{V}(\Omega), \\ \mathbf{R}_i^* &= \sum_{j \in \mathcal{N}_i} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T \frac{(\mathbf{w}_i^* - \mathbf{w}_j^*)}{2} \\ &\quad + \sum_{j \in \mathcal{N}_i} \mathbf{d}_{ij}^* + \sum_{j \in \mathcal{N}_i} \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right]^T \mathbf{w}_i^* \quad \forall i \in \mathcal{V}(\partial\Omega), \\ \mathbf{R}_i^* &= \sum_{j \in \mathcal{N}_i} \left[\frac{\partial(\mathbf{f}_i \cdot \mathbf{n}_{ij})}{\partial \mathbf{w}_i} \right]^T \frac{(\mathbf{w}_i^* - \mathbf{w}_j^*)}{2} \\ &\quad + \sum_{j \in \mathcal{N}_i} \mathbf{d}_{ij}^* + \sum_{j \in \mathcal{N}_i} \left[\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} \right]^T \mathbf{w}_i^* - \mathbf{g}_i^* \quad \forall i \in \mathcal{V}(\partial\Omega^o), \end{aligned} \quad (72)$$

where $\mathcal{V}(\partial\Omega^o)$ is the set of nodes at which the pressure P_e is measured, according to the definition of the BLE (19)–(22). The right-hand-side of equation (60), is included in the residuals

$$\mathbf{g}_i^* = \left(\left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{w}} \right)^* \mathbf{Q}^* \right)_i \quad (73)$$

The adjoint \mathbf{d}^* of the artificial dissipation fluxes \mathbf{d} (15) are obtained by freezing the artificial viscosities [2], that is, the differentiation of ϵ_2^{ij} and ϵ_4^i with respect to \mathbf{w}_h is assumed to give terms which can be neglected. This assumption yields that

$$\mathbf{d}_{ij}^* = \epsilon_2^{ij} (\mathbf{w}_i^* - \mathbf{w}_j^*) + (\epsilon_i \nabla^2 \mathbf{w}_i^* - \epsilon_j \nabla^2 \mathbf{w}_j^*). \quad (74)$$

However, the resulting truncation error may not be negligible in the computation of the gradient ∇J . This is studied later in this report through

numerical tests. A similar freezing of the coefficients in the farfield boundary conditions yields the following expression for the Jacobian of the farfield flux:

$$\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} = \frac{\partial(\mathbf{f} \cdot \mathbf{n}_i)}{\partial \mathbf{v}_i} \mathbf{L}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \mathbf{H}(\lambda_i) \mathbf{L}^{-1}(\hat{\mathbf{n}}_i, \mathbf{v}_\infty) \frac{d\mathbf{v}_i}{d\mathbf{w}_i}. \quad (75)$$

The Jacobian of the Euler wall flux function is expressed as follows

$$\frac{\partial(\mathbf{f}_i^{\text{bc}} \cdot \mathbf{n}_i)}{\partial \mathbf{w}_i} = (\gamma - 1) \begin{pmatrix} \frac{1}{2} |\mathbf{u}_i|^2 \\ -\mathbf{u}_i \\ 1 \end{pmatrix}. \quad (76)$$

3 Implementation issues

3.1 Mesh displacements

In the current study an explicit affine mapping is used to smoothly propagate changes in the geometry to the entire mesh. It is formulated as

$$\mathbf{X}^k = \mathbf{X}^0 + \mathbf{L} \mathbf{y}^k. \quad (77)$$

where $\mathbf{y}^k \in \mathbb{R}^n$ is the vector of node displacements on the airfoil, for the design number k , $\mathbf{X}^k \in \mathbb{R}^{2N}$ is the vector of all nodal coordinates (N is the total number of nodes in the grid), $\mathbf{L} \in \mathbb{R}^{2N \times n}$ is a constant coefficients matrix, and, \mathbf{X}^0 is the reference mesh, defined by its vector of nodal coordinates. Given the gradient ∇J_X , of a functional J_X such as defined in Table 1, the gradient of the functional $J_y(\mathbf{y}) \equiv J_X(\mathbf{X}(\mathbf{y}))$ is obtained by a matrix–vector product

$$\nabla J_y = \mathbf{L}^T \nabla J_X. \quad (78)$$

The definition of \mathbf{L} was possible because the meshes we used here are issued from structured grids. More general schemes are needed when using general unstructured meshes [7].

3.2 Parameterization of displacements and constraints

In shape optimization, the combination of gradient-methods and piecewise polynomial interpolations, such as B-splines, may induce oscillations in the shapes, as investigated in the reference [17]. In the current approach, smooth shapes are obtained, together with geometric constraints, by taking the vectors of displacements \mathbf{y} that are solution of a minimization problem [2] of the form

$$\mathbf{y} = \begin{cases} \min_{\mathbf{v} \in \mathbb{R}^n} \frac{1}{2} \mathbf{v}^T \mathbf{A}_s \mathbf{v} - \mathbf{v}^T \mathbf{M}_s \mathbf{a}, \\ \mathbf{C}^T \mathbf{y} = \mathbf{b} \end{cases}, \quad (79)$$

where A_s is the stiffness matrix associated with the Laplace operator, M_s is a mass matrix, C is a matrix whose rows are the gradients of constraints imposed on the displacements (in $\mathbb{R}^{n \times m}$) and \mathbf{b} is the vector of values imposed to the constraints (in \mathbb{R}^m). The solution \mathbf{y} to the above system is the vector of displacements, which, according to the norm defined by the stiffness matrix A_s , is the closest to the solution of the discretized Laplace equation defined by

$$A_s \tilde{\mathbf{y}} = M_s \mathbf{a}, \quad (80)$$

and that fulfills exactly the constraints

$$C^T \mathbf{y} = \mathbf{b}. \quad (81)$$

Such a parameterization implies that the controls are the vector \mathbf{a} , right side of the Laplace equation (80), and the vector \mathbf{b} , right side of the constraints relations (81). Therefore, from the gradient with respect to the displacements ∇J_y , that is eventually obtained by (78) it is needed to calculate a gradient with respect to $\{\mathbf{a}, \mathbf{b}\}$, the control variables in our method. This can be achieved by solving an adjoint problem of the form [2]

$$\begin{pmatrix} A_s^T & -C \\ -C^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y}^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} \nabla J_y \\ \mathbf{0} \end{pmatrix}, \quad (82)$$

from which it holds that

$$\nabla J_a = M_s^T \mathbf{y}^* \quad \text{and} \quad \nabla J_b = -\lambda^*. \quad (83)$$

3.3 Optimization algorithm

All numerical tests carried out here are formulated so that aerodynamic constraints (lift and pitch) are integrated to the objective function via a simple penalization technique. Geometrical constraints (volume and fixed domain) can be treated as simple bounds constraints via the parameterization previously discussed (see §3.2). However, the geometrical constraints are equality constraints, so that the right hand side, \mathbf{b} in (79), will be a constant vector. The only control parameter used in our applications is therefore the vector \mathbf{a} , expression (79).

The optimization algorithm is the limited memory quasi-Newton method developed by Byrd et al. [11]. It is based on a limited memory BFGS approximation of the Hessian matrix of the functional f , and a gradient projection method is used to account for bounds on the data which makes it suitable for large scale problems where only the gradient of f is available.

3.4 Solution procedure

A simple chart of the order in which the state and adjoint equations are solved and gradients are evaluated in order to perform optimal NLF is given

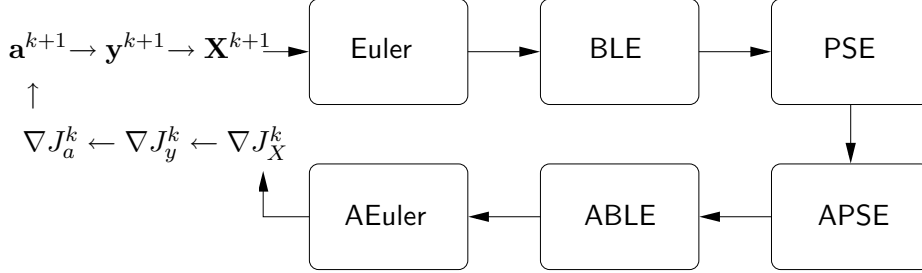


Figure 2: Flow chart for the case of minimizing the disturbance kinetic energy using the parameter \mathbf{a} to control the shape of geometry.

in figure 2. There, k denotes the iteration number and the procedure is as follows:

1. For $k = 1$, we start with an initial Euler mesh \mathbf{X}^0
2. The Euler, BLE and PSE are solved in the given order
3. The objective function J^k is evaluated
4. The adjoint equations, APSE, ABLE and AEuler are solved
5. The gradients ∇J_X^k , ∇J_y^k and ∇J_a^k are evaluated in the given order
6. A new control parameter \mathbf{a}^{k+1} is calculated² using the gradient ∇J_a^k
7. If $k > 1$, check convergence: If $|(J^{k+1} - J^k)/J^k| < \epsilon$, else³ continue
8. A new mesh \mathbf{X}^{k+1} is calculated from the new control parameter \mathbf{a}^{k+1}
9. Goto 2.

4 Numerical tests

The accuracy of the gradient ∇J_a (83) of the objective function is a critical issue in optimization. The first order necessary optimality condition is that the gradient of the objective function or, of the Lagrangian, is zero at an optimal design. Difficulties related to low accuracy, such as difficulty to find descent directions even far from the optimal design, are quite common. There are two possible causes of inaccuracies in our calculation of the gradient. As mentioned previously, the derivation of the adjoint of the discretized Euler equations makes use of an approximation as it does not linearize the coefficients of the 2nd order artificial dissipation. Effects of this

²In the computations shown here, the L-BFGS-B routine was used, normally requiring several functional and gradient evaluations in order to build up the approximative Hessian matrix.

³Several convergence criteria exist in the L-BFGS-B routine, [45]

approximation are investigated in §4.1. The adjoint equations of the BLE and PSE are derived from the continuous state equations. The effects of this method on the accuracy of the gradients of the type ∇J_w (for J_X defined as in Table 1) is investigated in §4.2.1. The calculation of ∇J_a is obtained by coupling the three systems of adjoint equations. The global accuracy is investigated in §4.2.2.

The accuracy of the gradient at a design point \mathbf{a} can be analyzed, comparing the value obtained from the solution of the adjoint equations, with the one estimated by finite differences as

$$(\nabla J_a)_k \approx \frac{J_a(\mathbf{a} + \epsilon_a \mathbf{e}_k) - J_a(\mathbf{a} - \epsilon_a \mathbf{e}_k)}{2\epsilon_a}, \quad (84)$$

where \mathbf{e}_k is the vector having component k equal to 1 and all other components being 0. Several calculations of $(\nabla J_a)_k$ is commonly performed, using different values of ϵ_a in order to find the best compromise between accuracy and rounding errors, the last being inherent to the finite difference method. The relative error between the gradient obtained by adjoint method $\nabla_{AD} J_a$ and the one approximated by finite-differences ∇J_a can be calculated as

$$\text{err}_{\nabla J_a} = \frac{\|\nabla_{AD} J_a - \nabla J_a\|}{\|\nabla J_a\|}, \quad (85)$$

where $\|\cdot\|$ denotes the norm in \mathbb{R}^n defined by the dot product.

These tests are performed on different C-type meshes of the RAE 2822. They are here denoted coarse, medium and fine, and the sizes are:

- Coarse: 3412 nodes with 112 nodes on the airfoil.
- Medium: 13352 nodes with 224 nodes on the airfoil.
- Fine: 52816 nodes with 448 nodes on the airfoil.

4.1 The inviscid case

Possible inaccuracies, due to the approximation made in the Euler adjoint equations, are expected to be independent from the forcing of the these equations. In our investigation the right-hand-side of the adjoint equation (60) is $(\partial \mathcal{A}_Q / \partial \mathbf{w}^*) \mathbf{Q}^*$ (see §2.4), identically equal to $\partial J_w / \partial \mathbf{w}$, according to the definition of J_w given in Table 1. It is therefore the same as usual aerodynamic optimization problems based on Euler flow analysis where J_w is usually one the functions defined by the wave drag, the lift or the moments coefficients. In order to avoid influences of possible errors when solving the BLE, the PSE and their adjoint equations, we investigate here the accuracy of the gradient of the wave drag (C_D), the lift (C_L) and the pitch moment coefficients (C_M).

The tests are carried out on the coarse grid, at three different design points (different Mach number and angle of attack). In the results shown

M_∞	$\text{err}_{\nabla C_L}$	$\text{err}_{\nabla C_D}$	$\text{err}_{\nabla C_M}$
0.754	1.9×10^{-2}	2.6×10^{-2}	1.6×10^{-2}
0.734	2.8×10^{-2}	3.8×10^{-2}	2.2×10^{-2}
0.68	5.5×10^{-3}	4.9×10^{-2}	4.6×10^{-3}

Table 2: Effect of the Mach number on the relative errors between the gradients calculated based on the adjoint equations and the ones calculated by finite differences, defined by (85), of the drag, lift and pitch moment coefficients.

M_∞	$\ \nabla C_L\ $	$\ \nabla C_D\ $	$\ \nabla C_M\ $
0.754	3.3×10^{-1}	3.0×10^{-2}	2.2×10^{-1}
0.734	3.2×10^{-1}	2.3×10^{-2}	2.0×10^{-1}
0.68	3.0×10^{-1}	2.4×10^{-3}	1.5×10^{-1}

Table 3: Effect of the Mach number on the norm of the gradient of the coefficients of lift (C_L), drag (C_D) and pitch (C_M). The gradients are calculated from the adjoint equations.

here, the value of ϵ_a in the finite difference approximation of the gradient $(\nabla J)_k$, is equal to 10^{-6} . However, various values in the interval $[10^{-4}, 10^{-8}]$ were tested without significant influence on the relative error. A summary of the results is given in Table 2. The relative error varies between 0.5% and 5% and clearly depends on the design point. Quite unexpected is that the error associated with the gradient of the drag coefficient (C_D) increases when reducing the Mach number whereas the influence of the 2nd order artificial dissipation would be expected to decrease as the influence of the shock decreases. Indeed, the influence of the shock on the flow solution is measured by the sensitivity of the wave drag and given in Table 3. Additional tests were carried out at Mach number 0.68 without the second artificial dissipation showing that the approximation in the derivation of the adjoint of these fluxes are causing the errors as, seen in Table 4.

4.2 The coupled inviscid-viscous case

The solutions of the adjoint of the stability and boundary layer equations are used as input to the adjoint of the Euler equations as well as the gradient assembly. From the adjoint of the boundary layer equations an expression for the gradient of the objective function with respect to the pressure distribution is obtained, expression (70). This is the coupling between the adjoint of the boundary layer equations and the adjoint of the Euler equations and is used to evaluate expression (73). The variation of the boundary layer

VIS2	$\text{err}_{\nabla C_L}$	$\text{err}_{\nabla C_D}$	$\text{err}_{\nabla C_M}$
1.	5.5×10^{-3}	4.9×10^{-2}	4.6×10^{-3}
0.	4.5×10^{-4}	3.4×10^{-3}	3.8×10^{-4}

Table 4: Effect of the 2nd order artificial viscosity fluxes on the relative errors between the gradients calculated from the adjoint equations and the ones calculated by finite differences for the drag, lift and pitch moment coefficients. The 2nd order artificial viscosity is active for $VIS2 = 1$ and inactive for $VIS2 = 0$.

and stability equations with respect to the node coordinates are additional terms in the assembly of the gradient, 61.

4.2.1 Gradient of objective function w.r.t. the pressure distribution

In the coupling between the inviscid and viscous solution, there are two issues which make a large impact on the accuracy of the gradients. The first is that a grid resolution commonly used to obtain results with the Euler equations, is too coarse to obtain converged results using the stability equations. A second issue is how the adjoint equations are derived. The adjoint Euler equations are derived from the discretized Euler equations, and its solution should not depend on the grid resolution. The adjoint of the boundary layer and stability equations on the other hand, are derived using the continuous approach, and the solution might therefore depend more on the grid resolution. As a consequence difficulties appear when increasing the accuracy.

Tests have been performed using the coarse, medium and fine grid when the free stream Mach number $M_\infty = 0.73$, Reynolds number $Re = 6.7 \times 10^6$, temperature $T_\infty = 300$ K and zero sweep angle. From the Euler solutions, the pressure distributions and coordinates of the upper side starting from the stagnation point have been used as input to the boundary layer and stability equations. The disturbance used in the stability calculations is a two dimensional wave with dimensional frequency $f^* = 15.5$ kHz. The objective function is evaluated as the disturbance kinetic energy, E_1 , integrated in a streamwise domain which is kept the same for the different grids used here.

In Figure 3 results are shown for a comparison between expression (70) and a finite difference approximation of the gradient of the objective function with respect to the pressure distribution. The latter is evaluated in the same way as expression (84), where the design \mathbf{a} has been replaced by the pressure P . Computations have been performed for different values of the finite-difference step ϵ_p , and these values are 10^{-4} , 10^{-5} , 10^{-6} , 10^{-7} . Only

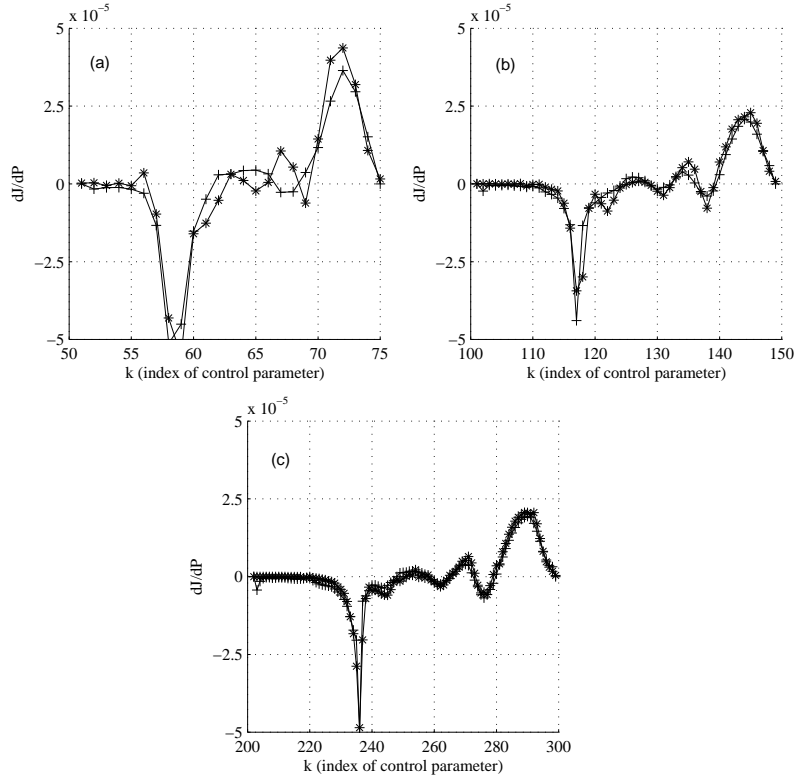


Figure 3: Gradient of objective function with respect to pressure distribution. Comparison between finite-difference approximation (solid-stars) and adjoint solution (solid-plus) computed on (a) coarse, (b) medium, and (c) fine grid.

the results for $\epsilon_p = 10^{-5}$ are shown. A first thing to note is that the gradient evaluated using the finite-difference approximation appears to converge as the grid resolution increases. This is due both to the solution of the pressure distribution converging in the Euler computation, and the solution of the stability equations converging as the grid is refined. Secondly it can be seen that the difference between gradients obtained from the finite-difference approximation and those obtained from the solution of the adjoint equations, decreases as the grid is refined. This is due to the continuous approach used to derive the adjoint equations.

The adjoint of the boundary layer and stability equations used here have been used in other problems concerning gradient evaluations for the purpose of optimal control problems [35, 36, 37]. There, different grid resolutions expressed as a step length in the local Reynolds number ΔRe , where $Re = \sqrt{U_e x / \nu_e}$, were tested to see the effect on the gradient accuracy. In

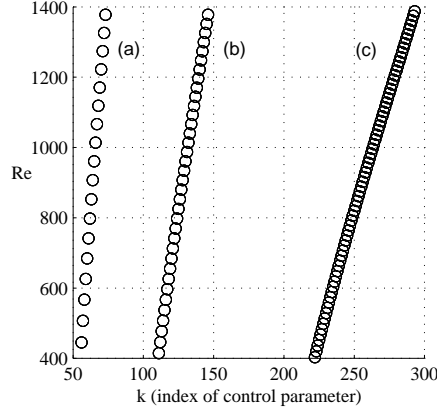


Figure 4: Local Reynolds number as a function of the control parameter k (a) coarse grid, (b), medium grid, (c) fine grid.

[35], it was shown that a step length of $\Delta Re = 20$ was needed to converge the physical result for a flat plate boundary layer with zero pressure gradient. Further, it was shown that a value of $\Delta Re = 10$ was needed to obtain a relative difference of 10^{-3} between the approximative finite difference calculation and the adjoint calculation, in the major part of the computational domain. In figure 4, the step length expressed as local Reynolds number has been plotted for the different grids used here. It can be seen that the step length is almost constant through out respective domain and the values are $\Delta Re = 50, 25$ and 13 for the coarse, medium and fine grids, respectively. Due to the variation of the pressure gradient along wing profiles, the step lengths found in the previous study of flat-plate boundary layers, might not be small enough to obtain the same convergence in the results and accuracy of the gradients in the case studied here. It should further be noted that the finest grid shown in figures 3 and 4, is computationally heavy for the Euler solver, and might not be reasonable to use in the case of optimization.

One option to increase the accuracy of the gradients is to use some interpolation technique for the calculation of the boundary layer, stability and corresponding adjoint equations, given the solution of the pressure distribution computed on a 'coarse' grid. Such an approach could also be favorable when three-dimensional flows are considered, and thus the computational effort increase considerably for the Euler equations. The outline of an interpolation approach can be made as follows: the pressure distribution from the Euler solution is interpolated on a grid with higher resolution, the boundary layer, stability and corresponding adjoint equations are solved on the new grid, the gradient ∇J_P is evaluated and then interpolated back on the original grid. Here, decisions need to be made on how to produce the new grid distribution, i. e. the distribution of the coordinate x/c describing the

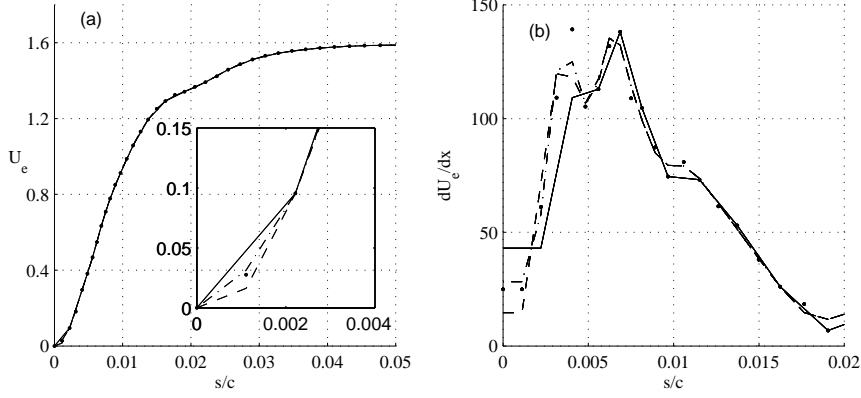


Figure 5: Comparison of U_e and $\partial U_e / \partial x^1$ between original medium grid (solid), and interpolated values using quadratic polynomial on half grid (dot), cubic splines on half grid (dash), and cubic splines on whole grid (dash-dot). The refined grids are obtained by inserting one additional point in between old grid points.

geometry, and further which interpolation scheme to use, in order to obtain y/c and P_e . One approach to refine the grid is to insert additional node points in between the old ones. This is a good approach if the original mesh has a uniform distribution (see figure 4). However, if this is not the case, the additional nodes might create locally small step sizes which is unfavorable for the stability equations. In such a case, a new distribution of desired quality must be made. This has not been considered so far. An advantage of inserting additional nodes in between the old ones, is that the function value (P_e) of the original nodes can be used in the chosen interpolation scheme. As the value of U_e plays an important role, and further as its value is zero at the stagnation point which does not change if the grid is refined, we have chosen to interpolate U_e instead of P_e . In Figure 5, results of U_e and $\partial U_e / \partial x^1$ can be seen using different interpolation schemes. We have used the medium grid shown earlier and the new grid is obtained by inserting an additional point in between the old grid points, i. e. the new grid has twice as many points as the original one. Two calculations have been performed when interpolation has been made using only the upper part of the wing, i. e. from the stagnation point to the trailing edge. The interpolation schemes are a piecewise quadratic polynomial and cubic splines. In addition a cubic spline interpolation has been made when the whole grid was used. In Figure 5a, results are shown of U_e close to the leading edge. A close up around the leading edge show that the magnitude of U_e of both cases where only the upper part of the wing was used in the interpolation, is larger at the first interpolation point. The streamwise derivative of the free stream velocity corresponding to the cases in Figure 5a are shown in Figure 5b.

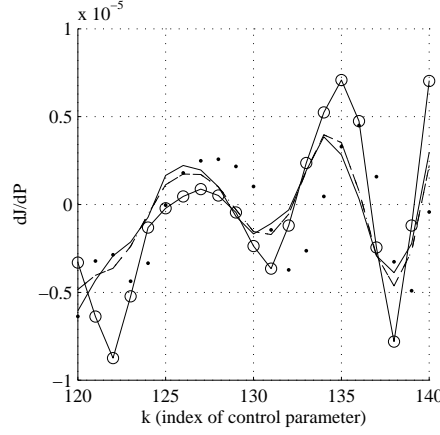


Figure 6: Comparison of the gradient $\partial J / \partial P_e$ computed from the solution of the adjoint equations using the mean flows corresponding to the cases in figure 5. A comparison is made with the finite difference approximation of the gradient (solid-circle).

The respective interpolated mean flows were then used to compute gradient ∇J_P by solving the PSE, APSE and ABLE. In Figure 6 a comparison is made between gradient computed using the finite difference approximation, and those obtained using the adjoint solutions. The gradient computed by solving the adjoint equations on the original grid is included for comparison. There, it is seen that the gradient computed using the piecewise quadratic polynomial interpolation technique oscillates and deviates most from the finite difference approximation. The cases of using the cubic splines cannot be distinguished from one another, and in comparison with the gradient computed on the original grid using the adjoint equations, a small improvement has been obtained. In the results of shape optimization for transition delay shown in this report, no interpolation technique has been used.

4.2.2 Gradient of objective function w.r.t. the shape parameterization

In the previous sections, tests have been performed to assess the accuracies of the inviscid (§4.1), and viscous parts (§4.2.1) separately. In this section, we investigate the accuracy of the gradient ∇J_a (83) which will be used in the optimization procedure, when the objective function is based on a measure of the disturbance kinetic energy. The reference case, denoted FD, is computed using the finite difference approximation of the gradient given by expression (84), which is evaluated solving consecutively the Euler, BLE and PSE. The objective function in all tests performed here is given as the disturbance kinetic energy of a single disturbance integrated in a

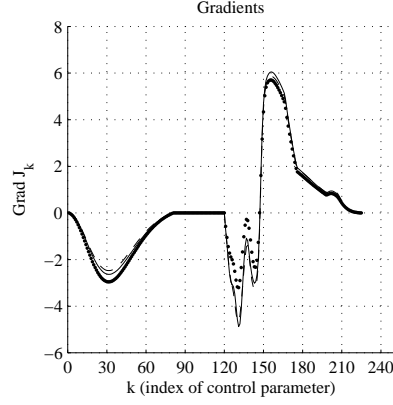


Figure 7: $M_\infty = 0.734$ - Gradients of the objective function (Disturbance kinetic energy) with respect to the optimization parameters (parameters that control the shape of the airfoil). The curves show FD (solid), ADJ1 (dot), ADJ2 (dash). The error between the FD and respective adjoint solution are: FD-ADJ1 (17.2%), FD-ADJ2 (4.95%), and FD-ADJ3 (4.85%, not shown).

defined streamwise region, expression (29) with $K=1$. The gradient ∇J_a is calculated from the gradient ∇J_X (61) by variables transformations, from the nodal coordinates \mathbf{X} to the displacements of the shape \mathbf{y} , according to (78), and, from the displacements of the shape \mathbf{y} to the right side of the Laplace equation \mathbf{a} , which requires to solve (82) and to apply (83). The gradient ∇J_X is given by expression (61)

$$\nabla J_X = \underbrace{\frac{\partial J_q}{\partial \mathbf{X}} - \left(\frac{\partial \mathcal{A}_q}{\partial \mathbf{X}} \right)^* \mathbf{q}^* + \left(\frac{\partial \mathcal{A}_Q}{\partial \mathbf{X}} \right)^* \mathbf{Q}^*}_{cont.} - \underbrace{\left(\frac{\partial \mathcal{A}_w}{\partial \mathbf{X}} \right)^* \mathbf{w}^*}_{discr.}.$$

As outlined in §3, it is evaluated from three systems of adjoint equations which are solved in the following order

$$\underbrace{APSE \rightarrow ABL E}_{cont.} \rightarrow \underbrace{\text{adjoint Euler}}_{discr.},$$

where each calculation depends on the solution of the previous one. The APSE and ABLE are derived using the continuous approach, while the adjoint Euler is derived using the discrete approach. This is above denoted *cont.*, and *discr.*, respectively. The latter has in the previous sections shown to be more accurate on the grids used here. In this section we therefore investigate the influence of the solution of the adjoint Euler, and the solution of the APSE and ABLE, on the accuracy of the gradient which will be used in the optimization. This is made by comparing the accuracy of the gradient computed from the solution of all adjoint equations with the one

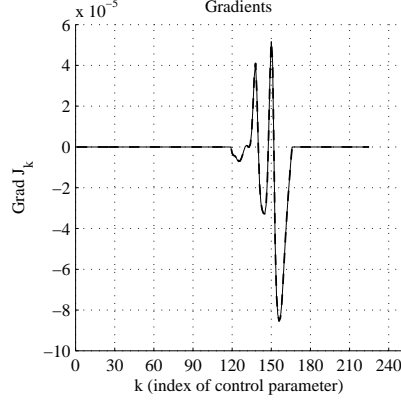


Figure 8: $M = 0.68$ - Gradients of the objective function (Disturbance kinetic energy) with respect to the optimization parameters (parameters that control the shape of the airfoil). The curves show FD (solid), ADJ2 (dash). The error is: FD-ADJ2 (0.02%)

evaluated from the solution of the adjoint Euler in which the right hand side, see expression (60), is approximated by finite differences. The right hand side is evaluated from expression (70) ($\partial J/\partial P$) when the solution of the APSE and ABLE are used. The finite difference approximation used here is the same that was used in §4.2.1. In addition, the influence of including the geometrical terms in the gradient evaluation from the solution of the PSE, APSE and ABLE (denoted *cont.*) is investigated.

The different cases are summarized below:

- FD: finite difference approximation of the gradient given by expression (84), computed solving the Euler, BLE and PSE.
- ADJ1: gradient evaluated from the solution of the adjoint Euler, ABLE, and APSE
- ADJ2: gradient evaluated from the solution of the adjoint Euler. The right hand side of the adjoint Euler, evaluated from $\partial J/\partial P$, is approximated by finite differences computed by BLE and PSE.
- ADJ3: gradient evaluated from the solution of the adjoint Euler, and by a finite difference approximation of the terms from the BLE and PSE. The right hand side of the adjoint Euler, evaluated from $\partial J/\partial P$, is approximated by finite differences computed by BLE and PSE.

The relative difference between the FD and respective gradient evaluated from the solution of the adjoint equations, is calculated using expression (85). In the first test we consider the medium grid when the Mach number $M_\infty = 0.734$. A part of the geometry around the leading edge is kept fixed, and a comparison is made between the FD and respective adjoint solution

outlined in the summary above. The gradients are plotted as functions of the index of the surface nodes, where index 224 denote the trailing edge. The fixed region around the leading edge is given between indexes 80 and 119. The largest error, 17.2%, is found in the comparison between the FD and the complete adjoint solution, ADJ1. When the right hand side of the adjoint Euler equations is approximated by finite differences, the error is reduced to 4.95%. Including the geometrical terms obtained from the BLE and PSE in the gradient evaluation (ADJ3) only reduces the error by 0.1% compared to the previous case (ADJ2).

The influence of including the second artificial viscosity on the gradient accuracy was shown in (§4.1) for the inviscid flow equations. As the influence is expected to decrease in the absence of a shock, a test was performed also here. We consider the flow at Mach number 0.68, and the surface is kept fixed everywhere, except for the region where the objective function is evaluated. In Figure 8, a comparison is made between the FD and ADJ2. The error in this case is 0.02%.

5 Optimization results

5.1 Description of the cases

Following the study performed on the accuracy of the sensitivities obtained using the continuous adjoint, §4.2, we chose to perform the optimization on the medium grid. Viscous calculations (RANS) are also carried out with EDGE⁴ [13] prior to and after some of the optimization tests in order to compare the N-factors based on the pressure distribution obtained from the viscous calculation with those that are computed using the Euler pressure distribution. These calculations are also used to compare the viscous drag between the initial and the final optimized design. The C-type grid for RANS calculations has the size:

- Medium 'RANS' mesh: 22088 nodes with 224 nodes on the airfoil.

In a first series of tests the objective is to reduce the disturbances kinetic energy. The only constraints are geometrical and imposed using the parameterization given in §3.2. The objective function to minimize is the total disturbance kinetic energy of a single disturbance, E_1 from expression (29).

There are several reasons for imposing geometrical constraints. A constant volume is intuitively a way to account for other industrial constraints such as having a minimum fuel tank capacity in the wings or a maximum weight of the material structure. The displacements of the nodes should not be allowed to be constant over all nodes, which would mean a translation of the wing. To remove this singularity one point should remain fixed, and

⁴The turbulence model used is the EARSM by [44] and the $k - \omega$ model.

our choice is the trailing edge. In addition, we chose to fix a region of the airfoil around the leading edge in order to prevent changes in the position of the stagnation point. To summarize, the geometrical constraints are:

- Constant volume,
- Constant position of the trailing edge,
- Fixed region around the leading edge (between 0 of the chord length and X_{ms} given in Table 5).

In order to test if the optimization can account for the usual aerodynamic requirements, two additional tests are carried out with a modified objective function, denoted J_C . The aim is to simultaneously reduce the disturbance kinetic energy and the wave drag, and in addition penalize changes in the coefficients of lift, and pitch-moment. The geometrical constraints are identical to the first type of optimization which is described above. The modified objective function is given as

$$J_C = \lambda_U E_1 + \lambda_D C_D + \frac{1}{2} \lambda_L (C_L - C_L^0)^2 + \frac{1}{2} \lambda_M (C_M - C_M^0)^2, \quad (86)$$

where E_1 is the functional from expression (29), C_D , C_L and C_M are the drag, lift, and pitch moment coefficients, respectively

$$\begin{aligned} C_D &= \sum_{i \in \mathcal{V}(\partial\Omega_w)} \frac{p_i \mathbf{n}_i \cdot \mathbf{d}_D}{\frac{1}{2} \rho_\infty \mathbf{v}_\infty^2 S_{\text{ref}}}, \\ C_L &= \sum_{i \in \mathcal{V}(\partial\Omega_w)} \frac{p_i \mathbf{n}_i \cdot \mathbf{d}_L}{\frac{1}{2} \rho_\infty \mathbf{v}_\infty^2 S_{\text{ref}}}, \\ C_M &= \sum_{i \in \mathcal{V}(\partial\Omega_w)} \frac{p_i \mathbf{d}_M \cdot (\mathbf{x}_i - \mathbf{O}_{\text{ref}}) \times \mathbf{n}_i}{\frac{1}{2} \rho_\infty \mathbf{v}_\infty^2 S_{\text{ref}} L_{\text{ref}}}. \end{aligned} \quad (87)$$

where \mathbf{d}_D is a unit vector in the direction of the farfield velocity, $\mathbf{d}_D = -\mathbf{v}_\infty / |\mathbf{v}_\infty|$, \mathbf{d}_L is a unit vector orthogonal to \mathbf{d}_D and, \mathbf{d}_M is a unit vector orthogonal to \mathbf{d}_D and \mathbf{d}_L .

The values C_L^0 and C_M^0 are the lift and the pitch moment coefficients for the initial design. In expression (86), we take the square of the deviation of lift and pitch with respect to the initial design in order to penalize both an increase and decrease during the optimization. The real numbers $\{\lambda_U, \lambda_D, \lambda_L, \lambda_M\}$ are scaling factors calculated as

$$\lambda_U = \frac{1}{10E_1^0}, \quad \lambda_D = \frac{1}{10C_D^0}, \quad \lambda_L = \frac{10}{(C_L^0)^2}, \quad \lambda_M = \frac{10}{(C_M^0)^2}. \quad (88)$$

The tests are summarized in Table 5. For the case T21, without second

Case	Objective	M_∞	Re_∞	f^* [kHz]	β^* [m^{-1}]	X_{ms}	X_{me}
T11	E_1	0.734	$1.7 \cdot 10^7$	15.5	0	0.043	0.45
T12	E_1	0.734	$6.5 \cdot 10^6$	11	500	0.043	0.45
T21	E_1	0.68	$1.6 \cdot 10^7$	16.5	0	0.039	0.45
T31	J_C	0.734	$1.7 \cdot 10^7$	15.5	0	0.043	0.45
T32	J_C	0.734	$6.5 \cdot 10^6$	11	500	0.043	0.45

Table 5: Optimization tests description.

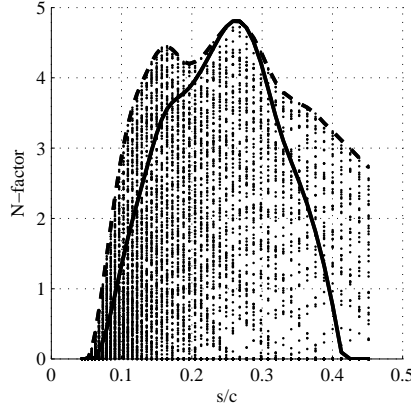


Figure 9: N-factor values for 165 modes (dots) with dimensional frequency $f^* = [5, 20]$ kHz ($\Delta f^* = 1$ kHz), spanwise wavenumber $\beta^* = [0, 2500]$ m^{-1} ($\Delta \beta^* = 250$ m^{-1}). From these values is the mode used in the optimization (solid) chosen, and the envelope of envelopes (EoE) (dash) calculated. The flow in this case is characterized by $Re_\infty = 6.5 \cdot 10^6$, $M_\infty = 0.734$, $\alpha = 2.1875$ degrees.

order artificial viscosity (VIS2=0) was used in order to get better accuracy (see Figure 8). The thermodynamical properties for the different cases correspond to two different altitudes such that the TX1-cases are given at 0 meter ASL, and the TX2-cases are given at 9600 meter ASL.

5.2 Analysis of the disturbance growth

The objective function in the results shown here, is given as the total disturbance kinetic energy of a single disturbance, expression (29) in which $K = 1$. A stability analysis of a large number of modes with different frequencies f^* , and spanwise wave numbers β^* corresponding to different wave angles, is performed prior to each optimization case, on the original design and the chosen flow conditions. The wave angle is defined as the angle

between the wave number vector \mathbf{k} and the inviscid streamline. The corresponding N -factors are calculated from these results, and the mode chosen to be used in the optimization which has the largest N -factor value with respect to all other modes, in the computational domain. The reason of this particular choice is that it has been shown in previous studies on optimal control, [37, 36], that a control that successfully decrease the growth of a single disturbance also have a damping effect on other instability waves of the same type. It is common in transition prediction, to compute the envelope or envelopes (EoE) of the N -factor curves (i. e. envelope over both frequency and spanwise wave number). Transition is then assumed to occur at the position where the EoE curve first attains an empirically determined value. This curve also serves as a measure of the efficiency of a control or design, computed by minimizing a single disturbance, on a large number of disturbances.

Results of the analysis discussed here is shown in Figure 9. The design is the medium mesh with a free stream Mach number $M_\infty = 0.734$, Reynolds number $Re_\infty = 6.5 \cdot 10^6$, and angle of attack $\alpha = 2.1875$ degrees. A total of 165 modes have been analyzed with dimensional frequency $f^* = [5, 20]$ kHz ($\Delta f^* = 1$ kHz), and spanwise wave number $\beta^* = [0, 2500]$ m^{-1} ($\Delta \beta^* = 250$ m^{-1}). This choice of spanwise wave number corresponds to wave angles between zero and 85 degrees. The corresponding N -factor values of all modes are given by dots. The mode chosen to be used in the optimization is given by the solid line and the EoE curve by the dash line. The values of f^* and β^* given here are used for all EoE analysis made in this report.

5.3 Reduction of disturbance kinetic energy under volume constraints

5.3.1 Cases T11 and T12

Results are shown here for the case of minimizing the disturbance kinetic energy of a single disturbance with the initial volume of the airfoil kept constant. Computations are performed for a given Mach number $M_\infty = 0.734$ and two different Reynolds numbers. The latter two correspond to 0, and 9600 meter ASL, and the cases are denoted T11 and T12, respectively.

The convergence history is given in Figure 10 for the T11 case. The objective function and gradient norm is given as a function of the iteration number. The optimization was stopped because the BLE could not converge for the design after the last iteration. This occurred as the changes in the geometry caused the shock wave to move upstream $x/c \approx 0.42$, into the domain in which the objective function is evaluated (between $x/c = 0.043$ and $x/c = 0.45$). This can be seen in Figure 11, where the pressure coefficient and geometry for the initial and final design are plotted.

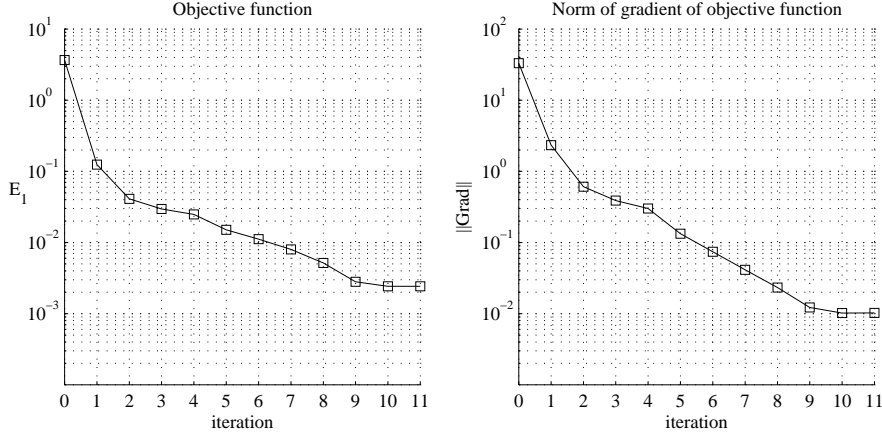


Figure 10: T11 - Objective function and gradient norm.

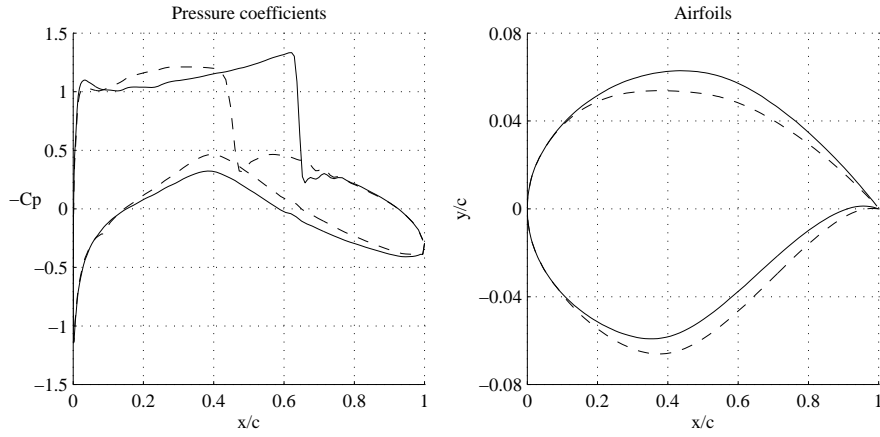


Figure 11: T11 - Pressure coefficients and shapes at initial (solid) and final design (dash).

At final design, the central upper part of the wing is thinner, measuring the thickness as the distance of a point on the airfoil to the chord. Therefore, because of the fixed region around the leading edge, the region between 4.3% of the chord length, from the leading edge, up to about 30% of the chord length, situated on the upper part, has a higher curvature at final design than at initial design. An increase of the curvature of a wall boundary is known to reduce the pressure in the fluid flow. This may be the effect that can be observed in Figure 11 where the pressure coefficient at final design has decreased ($-C_p$ is increased) in the region between 10% of the chord length, from the leading edge, up to about 30% of the chord length, in comparison to the initial design. In this way a pressure gradient is obtained that damps the growth of disturbances as it is explained below. However, the faster

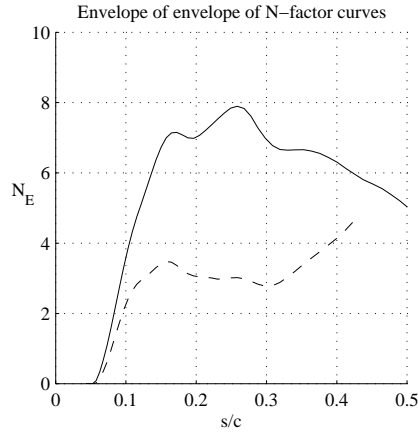


Figure 12: T11 - Envelope of envelopes of N -factor curves. Comparison between initial (solid) and final design (dash).

decrease of the pressure may be responsible for the shock moving upstream. Note that the deformation of the lower part of the wing is only due to the constraint that imposes a constant volume. The effect on the disturbance growth can be seen in Figure 12 where the EoE curves have been plotted for the initial and final design. A large damping of all modes has been achieved. This can be explained by the change in pressure gradient from adverse to favorable, in a large part of the region where the disturbances are amplified. The disturbance growth increases in this region due to the zero or weak adverse pressure gradient just upstream of the shock wave in the case of the final design.

The convergence history is given in Figure 13 for the lower Reynolds number case (T12). The decrease of the objective function is of two orders of magnitude, smaller than it is for the case T11, Figure 10, but the norm of the gradient is decreased by four order of magnitude, which is larger than the decrease achieved in case T11. The computation is here terminated as no further descent direction could be found. The magnitudes of the deformations of the airfoil are smaller compared to the case T11, shown in Figure 11, but a similar trend is observed. Pressure gradients that are favorable, to a decrease of the energy of the disturbances, may be caused by a local increase of the curvature on the upper part of the wing, between 4.3% and 30% of the chord length away from the leading edge. As a consequence, the shock is moved upstream.

The effect on the disturbance growth can be seen in Figure 15 where the EoE curves have been plotted for the initial and final design. A large decrease in disturbance growth is obtained using the optimized design, similar to the one found for the high Reynolds number case (T11). Also this is due to the change in pressure gradient from an adverse to favorable in

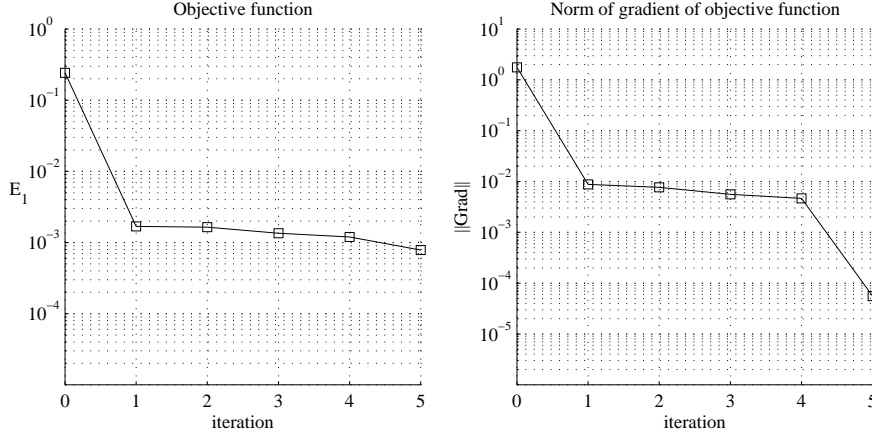


Figure 13: T12 - Objective function and gradient norm.

the upstream part of the domain where the disturbances become unstable. The shock wave has not moved as far upstream and the boundary layer and stability analysis can therefore be made further downstream, also on the optimized airfoil.

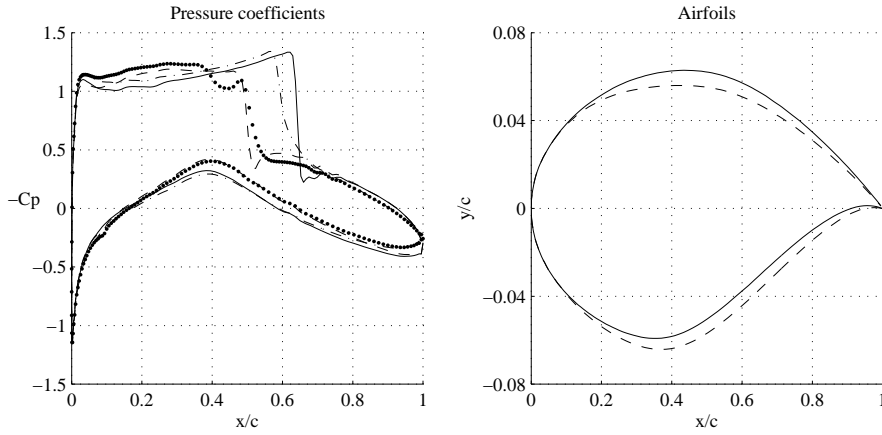


Figure 14: T12 - Pressure coefficients and shapes at initial design for Euler (solid) and RANS (dash-dot), and final design for Euler (dash) and RANS (dot).

An attempt has been made to use the N -factor results of the initial and final design in order to determine the respective transition location. These results have been used as input to RANS calculations of the initial and final design in order to evaluate the change in the viscous drag. Values of the drag, lift and pitch coefficients for the initial and optimized designs are summarized in Table 6. The viscous drag is reduced by 13 drag counts, and,

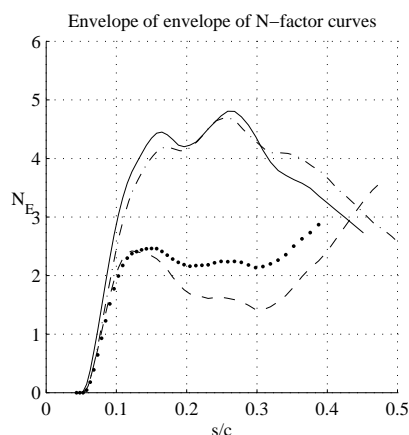


Figure 15: T12 - Envelope of envelopes of N -factor curves. Comparison between initial (solid) and final design (dash). A comparison is also made between the initial (dash-dot) and final (dot) design, when the pressure distribution is given by the solution of the Reynolds Averaged Navier Stokes equations.

as it could be expected from previous observations about the position of the shock, the wave drag is also decreased, by 40 drag counts⁵. However, this is a byproduct of the reduction of the disturbance energy. Large changes in the lift and the pitch are also observed. These by-effects are controlled by imposing constraints in the cases T31 and T32, see §5.4.

The results from the RANS calculations are in addition used to see the difference in N -factors computed using the pressure distributions from the Euler-, and RANS solutions. The transition position on the upper side of the initial design was taken as the streamwise position corresponding to the maximum value of the EoE curve of the N -factors computed using the pressure distribution from the Euler solution. The value is $s/c \approx 0.26$, see Figure 15. As the EoE curve of the final design was lower in magnitude compared to the initial one, the transition location of the final design was set as the downstream position of the computational domain of the boundary layer and stability analysis. The same transition position was used on the lower side, both for the initial and final design. It should be noted that the transition location for the initial design is not based on experimental results. A common reference for the RAE2822 airfoil is [12], in which the boundary layer was tripped at 3% chord in the experiments in order to have a well defined turbulent portion.

The pressure coefficients obtained from the solution of the RANS for the initial and final design is found in Figure 14. The largest difference

⁵One drag count is 10^{-4}

Coeff.	Case	RANS			Euler
		total	viscous	pressure	pressure
C_D	Initial	2.3×10^{-2}	4.8×10^{-3}	1.8×10^{-2}	1.3×10^{-2}
	T12 (Final)	1.6×10^{-2}	3.5×10^{-3}	1.3×10^{-2}	7.9×10^{-3}
	T32 (Final)	1.8×10^{-2}	3.7×10^{-3}	1.4×10^{-2}	8.3×10^{-3}
C_L	Initial	8.4×10^{-1}	-7.9×10^{-5}	8.4×10^{-1}	8.4×10^{-1}
	T12 (Final)	7.0×10^{-1}	-5×10^{-5}	7.0×10^{-1}	7.0×10^{-1}
	T32 (Final)	8.5×10^{-1}	-7×10^{-5}	8.5×10^{-1}	8.5×10^{-1}
C_M	Initial	3.2×10^{-1}	—	—	3.4×10^{-1}
	T12 (Final)	2.4×10^{-1}	—	—	2.6×10^{-1}
	T32 (Final)	3.1×10^{-1}	—	—	3.4×10^{-1}

Table 6: Summary of aerodynamic coefficients at initial and final design for T12 and T32, using Euler and RANS flow analysis.

compared to the Euler solution occur at the position of the shock wave. Upstream of this position however, the difference between the Euler and RANS solution is smaller, which can also be seen in Figure 15, where the EoE curves of the two cases are compared. In the comparison of the pressure coefficients between the Euler and RANS computations of the final design, one can note the difference in the region between $x/c \approx 0.4$ and $x/c \approx 0.5$. Results of the velocity field from the RANS calculation (not shown here), show that separation occurs in this region. As this can not be accounted for in the boundary layer equations used here, the EoE curve computed using the pressure distribution from the RANS calculation is not performed downstream of $x/c \approx 0.4$, see Figure 15.

5.3.2 Case T21

Results are shown here for the case of minimizing the disturbance kinetic energy of a single disturbance with the initial volume of the RAE2822 airfoil kept constant. Computations are performed for Mach number $M_\infty = 0.68$ at 0 meter ASL. The shock wave shown in the T11-, and T12 cases is not present in this case, see Figure 17. The convergence history is given in Figure 16. The objective function and the norm of the gradient are reduced of about three orders of magnitude. The pressure coefficients, and the geometries, at initial and final design, are found in Figure 17.

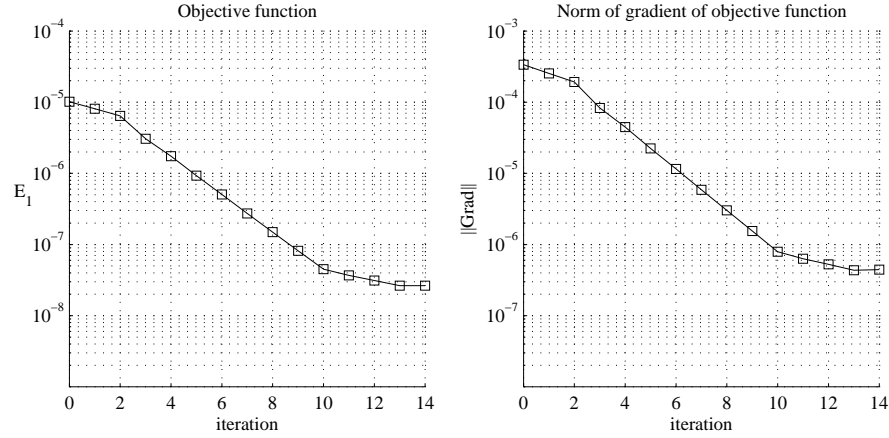


Figure 16: T21 - Objective function and norm of its gradient.

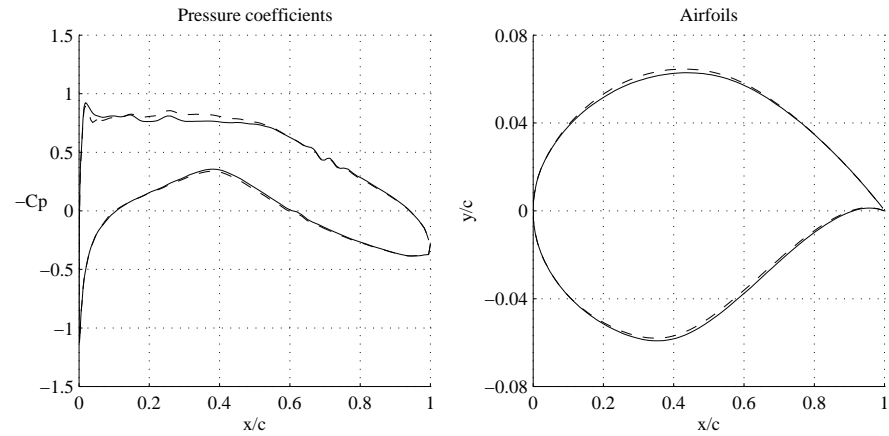


Figure 17: T21 - Pressure coefficients and shapes at initial (solid) and final design (dash).

Alike the previous cases, the changes in the geometry of the upper part of the airfoil influence the pressure distribution in a way that is favorable to decrease of the disturbance energy. However, the deformation of the airfoil is of a different nature. Compared to the cases T11 and T12, the upper part of the airfoil is thicker at final design than at initial design, which creates a region of higher curvature at about 40% of the chord length away from the leading edge. This is further downstream compared to where it appeared in the cases T11 and T12. The effects of these deformations can be observed in changes of the pressure coefficient.

The difference between the test cases is better understood looking at the envelope of envelopes (EoE) curves of the N -factors. In case T11, the disturbances grow fastest in a region between $x/c = 0.2$ and $x/c = 0.3$, as

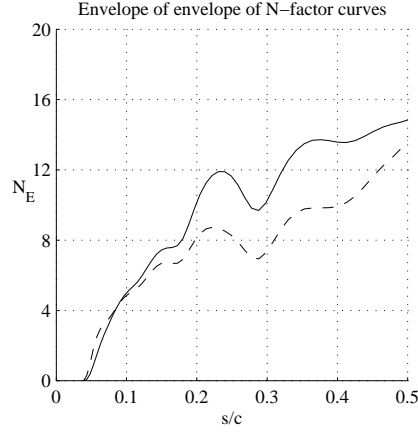


Figure 18: T21 - Envelope of envelopes of N -factor curves. Comparison between initial (solid) and final design (dash).

seen in Figure 12. In the case T21, the growth of disturbances continues increasing outside of the domain of integration of the objective function (29), that is for $x/c > 0.45$, Figure 18. As a consequence, in case T11, damping the growth of disturbances in the near region of the leading edge has a major effect, but in case T21 the damping may be favored as far downstream as $x/c = 0.45$. A decrease in disturbance amplification has been achieved for all modes in the major part of the computational domain. This is due to change in the pressure distribution from a weak adverse to a zero or weakly favorable in the region where the disturbances become unstable, see Figure 17. Note however, that in the upper most streamwise region, first an increase in the adverse pressure gradient occur. This is seen in the EoE curves which are actually larger between $x/c \approx 0.05$ and $x/c \approx 0.1$ compared to the initial design. Another thing to note is the “smoothing” of the pressure distribution in the region where the disturbances are amplified. The vanishing of these wiggles is also seen in the EoE curve plotted for the final design.

5.4 Reduction of disturbance kinetic energy and wave drag under volume constraints while penalizing lift and pitch

Results are shown here for the case of simultaneously minimizing the disturbance kinetic energy of a single disturbance, and the wave drag. The initial volume is kept constant during the optimization and changes in the lift, and pitch-moment coefficients with respect to the initial design are penalized, see expression (86). Computations are performed for a given Mach number $M_\infty = 0.734$ and two different Reynolds numbers. The latter two correspond to 0, and 9600 meter ASL, and the cases are denoted T31 and T32, respectively.

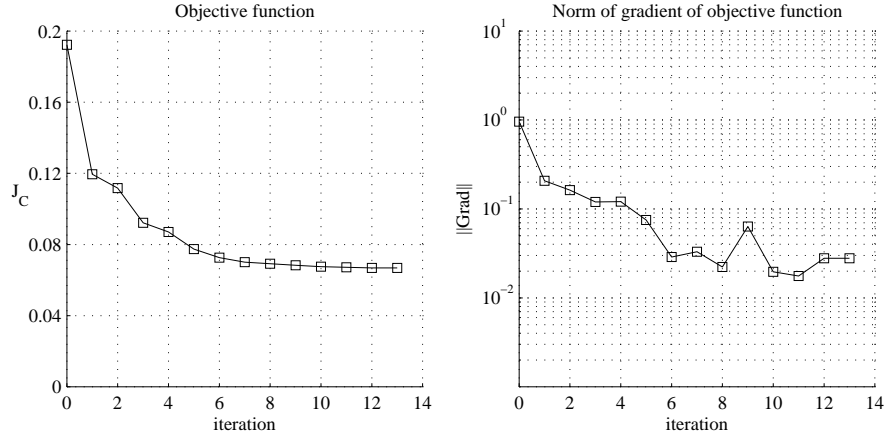


Figure 19: T31 - Objective function and gradient norm.

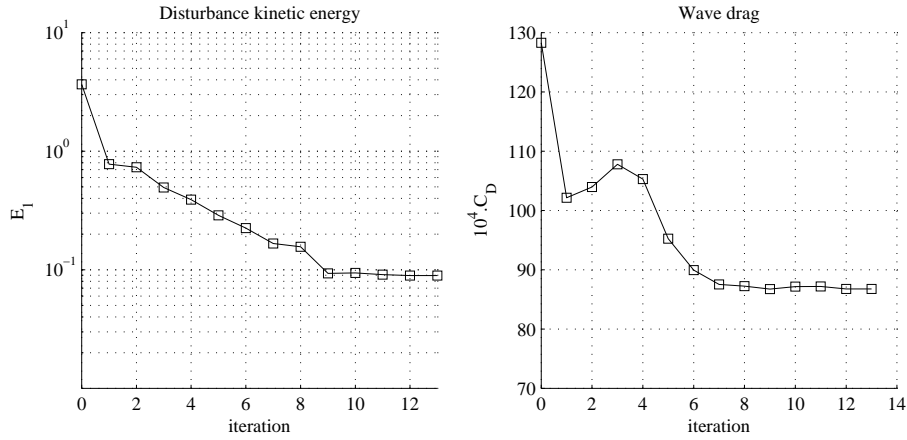


Figure 20: T31 - Disturbance kinetic energy and wave drag.

The objective function and gradient norm of case T31 are given as functions of the iteration number in Figure 19. The different components of the objective function are plotted in figures 20, and 21. The objective function is decreased in each step of the optimization even though the component of the wave drag is increased between iteration number 2 and 4, compared to iteration number 1. The reason is that in this interval, the deviation of lift-, and pitch moment coefficients is decreased. A reduction has been obtained at the last iteration, in both disturbance kinetic energy and wave drag, while the lift-, and pitch moment coefficients are kept within a few percent. Comparisons between the pressure coefficients, and geometries of the initial and final design are given in Figure 22. The change in pressure distribution occurs mainly on the upper side of the airfoil, where the shock wave

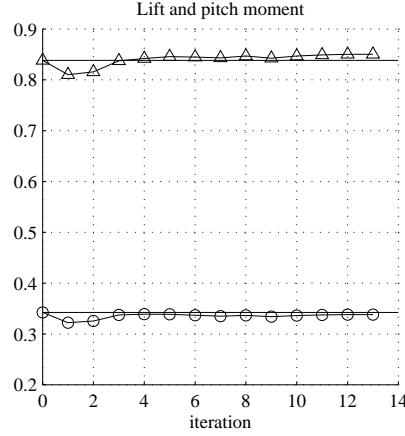


Figure 21: T31 - Lift (triangle-solid) and pitch moment (circle-solid) coefficients. The values at initial design are indicated at each step (solid).

has moved upstream and weakened. In comparison with T11, which has the same initial conditions, the displacement of the shock in T31 is smaller, but, the changes in T31 reflect the conservation of the pitch moment and lift coefficients.

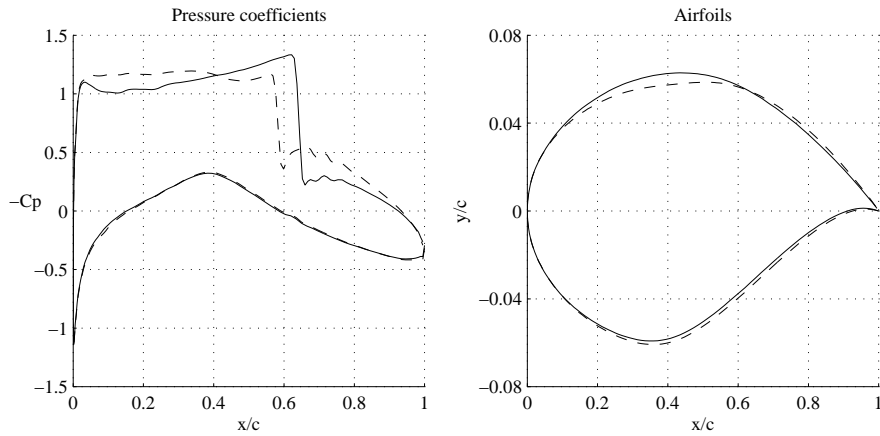


Figure 22: T31 - Pressure coefficients and shapes at initial (solid) and final design (dash).

The effect on the disturbance growth can be seen in Figure 23 where the EoE curves have been plotted for the initial and final design. A damping of the disturbance growth is obtained in a large part of the computational domain using the final design. It is clear looking at Figure 22, that the adverse pressure gradient of the initial design in the upstream region where the disturbances become unstable, has changed into a zero or weakly favorable

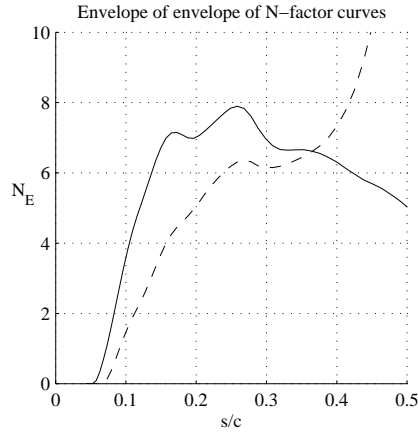


Figure 23: T31 - Envelope of envelopes of N -factor curves. Comparison between initial (solid) and final design (dash).

in the final design. Close to the shock wave of the final design, which has now moved further upstream, the flow is decelerated. This can be seen in the EoE curve where the value increases rapidly above that of the initial design.

The convergence history for the lower Reynolds number case (T32) is found in figures 24–26, and is similar to the one found for the case T31. The wave drag experiences an increase during a few optimization steps also here, while the deviation of lift, and pitch-moment coefficients decreases. In Figure 27, a comparison is made between the pressure coefficient and geometry of the initial and final design. In comparison with case T12, which has the same initial conditions, the displacement of the shock is smaller but the pressure distribution in T32 minimizes changes in the coefficients of lift and pitch moment. The EoE curves computed using the pressure distribution from the Euler solution are used, as in case T12, to set the transition locations in two RANS calculations. The transition position on the upper side of the initial design was taken as the streamwise position of the maximum value of the EoE curve of the N -factors computed using the pressure distribution from the Euler solution ($s/c \approx 0.26$, see Figure 28). The maximum value of the EoE curve computed for the final design is below the value found for the initial one. The transition location on the upper side of the final design is therefore set as the downstream position of the computational domain of the boundary layer equations. The same transition position was used both for the initial and final design on the lower side. The major differences in the pressure distribution between the Euler and RANS calculation occur in the region around the shock wave. Upstream of the shock the difference is smaller, both for the initial and final design. This can also be seen in the comparison of the EoE curves found in Figure 28.

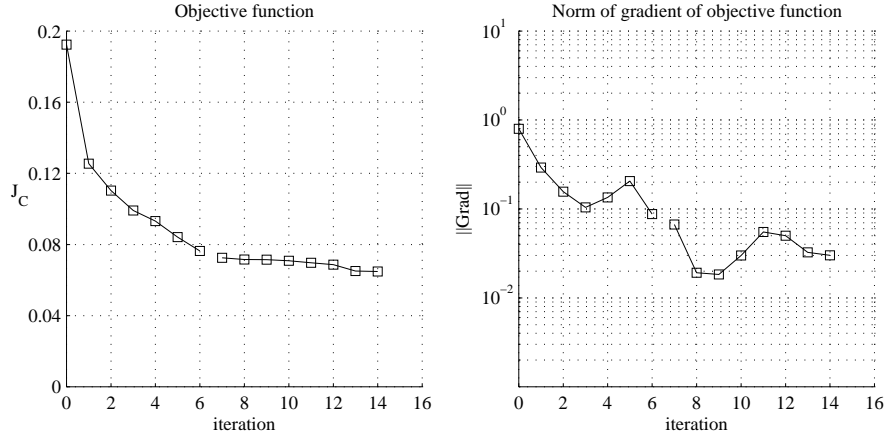


Figure 24: T32 - Objective function and norm of its gradient.

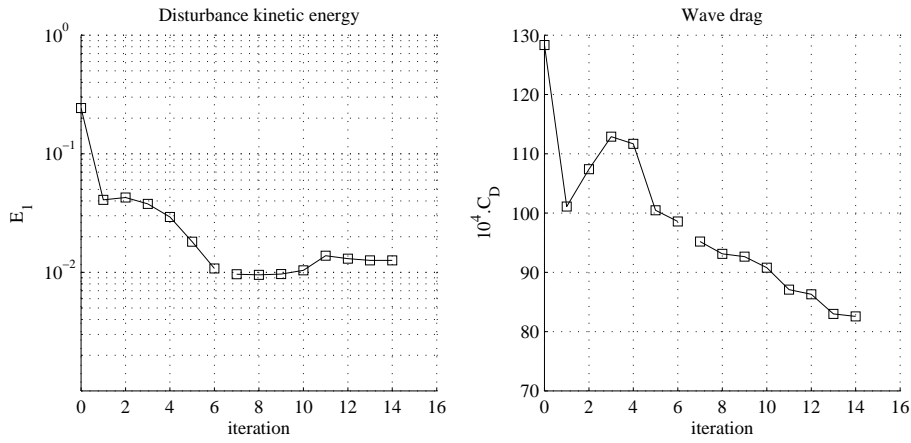


Figure 25: T32 - Disturbance kinetic energy and wave drag.

A decrease occur in the N -factor values in both results showing the EoE curves of the final design. This can be explained by the change in pressure gradient from adverse to zero or favorable, in a large part of the region where the disturbances are amplified.

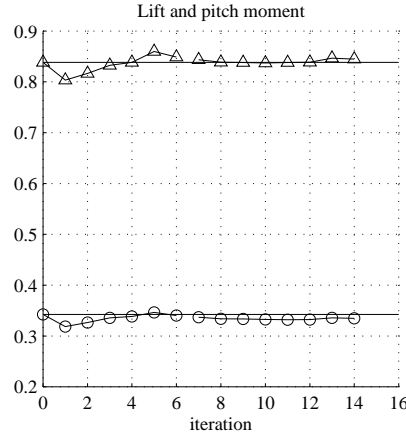


Figure 26: T32 - Lift (triangle-solid) and pitch moment (circle-solid) coefficients. The values at initial design are indicated at each step (solid).

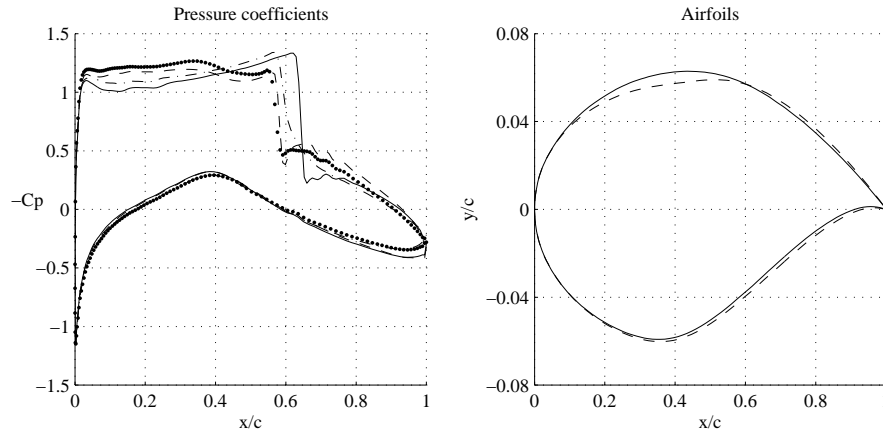


Figure 27: T32 - Pressure coefficients and shapes at initial design for Euler (solid) and RANS (dash-dot), and final design for Euler (dash) and RANS (dot).

6 Summary and discussion

Theory and results have been presented for an approach to perform shape optimization with the aim of transition delay, and thus a decrease of the viscous drag. The location of laminar-turbulent transition is analyzed using linear stability theory, in which perturbations with infinitely small amplitude are superimposed on the laminar mean flow. It is then assumed that transition will occur at the location where the total amplification of distur-

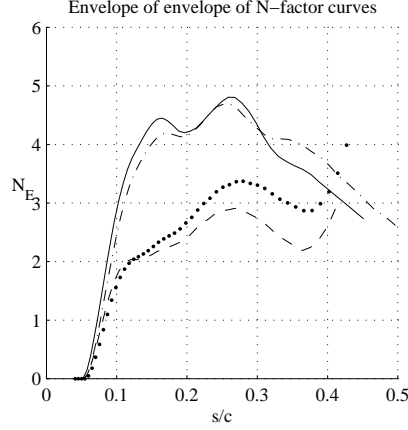


Figure 28: T32 - Envelope of envelopes of N -factor curves. Comparison between initial (solid) and final design (dash). A comparison is also made between the initial (dash-dot) and final (dot) design, when the pressure distribution is given by the solution of the Reynolds Averaged Navier Stokes equations.

bances, with respect to the first streamwise position where the disturbance starts to grow, attains an empirically determined value, whose logarithm is generally denoted by N . The inviscid flow is obtained by solving the Euler equations for compressible flows, and the viscous mean flow is obtained from the solution of the boundary layer equations for compressible flows on infinite swept wings. The evolution of convectively unstable disturbances is analyzed using the linear parabolized stability equations (PSE).

In the present approach, an iterative gradient based optimization procedure is used with the aim of minimizing an objective function based on the disturbance kinetic energy. Tests are carried out starting from the RAE2822 airfoil and are formulated to produce a reduction of the disturbance kinetic energy while maintaining a fixed volume, angle of attack, region around the leading edge and trailing edge position. Flow conditions include transonic and subsonic cases, with Reynolds number of 6.5, and 17 millions. In some cases the objective function is formulated to simultaneously reduce the wave drag and the disturbance kinetic energy while maintaining lift and pitch moment coefficients near their values at initial design. The normal displacements of the nodes on the airfoil are solution of a quadratic programming problem minimizing the variational form of the discrete Poisson problem and including linear constraints. Such a parameterization ensures smoothness of the geometries for each design generated by the quasi-Newton optimization algorithm [10] and enables to define complex sets of admissible shapes as needed when coupling the three state equations. It has been shown that the gradient of the objective function with respect to the design

variables can be evaluated from the solution of adjoint of the Euler, boundary layer and parabolized stability equations. Using the adjoint equations, as opposed to other perturbation techniques, constitutes an efficient way to evaluate functional gradients when the number of design variables is large compared to the number of objective functions.

The work presented here is an ongoing project with improvements to be made in order to increase the existing computational efficiency and accuracy. It is further possible to include additional physical modeling in order to generalize the technique for more complex flow situations. As discussed herein, there are issues related to grid resolution and the approach taken to derive the adjoint equations. From a physical point of view, in order to obtain a converged result, different grid resolutions might be required for the Euler, boundary layer and stability equations. As shown here, the medium RAE2822 grid has a resolution which is fine with respect to what is needed for the Euler solution and coarse with respect to the stability solution. However, further refinement would not give large improvements in the physical results. For the evaluation of the gradient of the disturbance kinetic energy with respect to the pressure distribution on the other hand, the resolution plays an important role, as the ABLE and APSE are derived using the continuous approach. For this reason effort must be put into either deriving these equations using the discrete approach or to improve on the interpolation technique shown here.

There are modifications which can be made in order to approach a more realistic situation. One is to include the disturbance growth also on the lower side of the wing in the objective function, as transition occurs on both sides of the wing. As the objective function of the total disturbance kinetic energy is given as the sum of K convectively unstable disturbances, test can also be made in order to evaluate this effect. Another extension, which can include both of the above, is to use multiple design points in the optimization. In such case the objective function is the sum of a chosen cost function at e. g. different Mach numbers and/or different disturbances.

With the approach taken here there are some limitations which could be overcome using additional physical modeling. As no iterative coupling exist between the pressure distribution and the thickness of the boundary layer, this constitutes an approximation. In addition, the boundary layer calculated here is assumed laminar, and the effect of the increased thickness of the turbulent boundary layer due to transition is not accounted for. Separation is another issue which is not taken into account, and can be important especially for applications with large angle of attack. Several studies have been made on this topic for the boundary layer and stability analysis and might be possible to include in the current project. Another option that should be tested in order to avoid large adverse pressure gradients which might cause separation in the downstream domain, is to minimize an objective function including both E_K and E_f . The idea to include also the disturbance kinetic

energy at a downstream position (E_f), is based on the knowledge that an adverse pressure gradient has a destabilizing effect on the disturbance growth. Instead of using the Euler equations an extension is to use the Reynolds Averaged Navier-Stokes equations (RANS). This development should include the adjoint of the RANS equations.

Even though the Euler and adjoint Euler equations can be derived and solved for complete three-dimensional flows, the boundary layer and stability equations used here are given for, at most, infinite swept wing flows. The absence of variation of the viscous mean flow in the spanwise direction in the BLE and PSE means that e. g. tapered wings can not be analyzed without further approximations. To proceed to fully three-dimensional cases, it therefore has to be decided what approximations to make, or if effort should be spent on solving the BLE and PSE for complete three-dimensional flows.

The delay of transition is a benefit if accounting for all other aerodynamic properties of the wing, which are the wave drag, lift and moments coefficients. Reducing the wave drag is as important as the delay of transition when optimizing the airfoil at cruise speed. The latter can be achieved by formulating the objective function as a weighted sum of the wave drag and the disturbance kinetic energy. Lift and pitch moment can also be maintained near to their value for the initial design by adding terms to the objective function that penalize variations of these coefficients. However, in real life applications the interest is to maintain the lift above a minimum level while fixing the moments to their value at the original design. Imposing bound constraints in this non linear optimization problem would require to use more advanced method than the penalization technique that has been used in this study.

Acknowledgment

The financial support of the EU project (ALTTA contract G4RD-CT-2000-00143) is gratefully acknowledged. This research has received support from the AEROSHAPE project funded by the European Commission, DG Research, under the GROWTH initiative (Project Ref: GRD1-1999-10752).

References

- [1] C. Airiau, A. Bottaro, S. Walther, and D. Legendre. A methodology for optimal laminar flow control: Application to the damping of Tollmien-Schlichting waves in a boundary layer. *Phys. Fluids*, 15:1131–1145, 2003.
- [2] O. Amoignon. Adjoint-based aerodynamic shape optimization, 2003. IT Licentiate theses 2003-012, Department of Information Technology,

Division of Scientific Computing, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden.

- [3] W.K. Anderson and D.L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.
- [4] D. Arnal. Boundary layer transition: Predictions based on linear theory. Special course on 'progress in transition modeling', March–April 1993, AGARD-R-793, 1993. 2-1–2-63.
- [5] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations. In *Special Course on Unstructured Methods for Advection Dominated Flows*, pages 6–1–6–61. AGARD Report 787, May 1991.
- [6] O. Baysal and K. Ghayour. Continuous adjoint sensitivities for optimization with general cost functionals on unstructured meshes. *AIAA Journal*, 39(1):48–55, 2001.
- [7] M. Berggren. Edge-based mesh movement strategies, 2003. Unpublished preprint, Sandia National Laboratories.
- [8] F. P. Bertolotti, T.H. Herbert, and S.P. Spalart. Linear and nonlinear stability of the Blasius boundary layer. *J. Fluid Mech.*, 242:441–474, 1992.
- [9] G.W. Burgreen, O. Baysal, and M.E. Elshaky. Improving the efficiency of aerodynamic shape optimization. *AIAA Journal*, 32(1):69–76, 1994.
- [10] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. Technical Report NAM-08, Northwestern University, Department of Electrical Engineering and Computer Science, Evanston IL 60208, 1994.
- [11] R.H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, 1995.
- [12] P.H. Cook, M.A. McDonald, and M.C.P. Firmin. Aerofoil RAE 2822 – pressure distributions, and boundary layer and wake measurements. Experimental data base for computer program assessment, AGARD-AR-138, 1979.
- [13] P. Eliasson. Edge, a Navier–Stokes solver, for unstructured grids. Technical Report FOI-R-0298-SE, Swedish Defence Research Agency, Stockholm, November 2001.

- [14]J. Elliot. *Aerodynamic based on the Euler and Navier-Stokes equations using unstructured grids*. PhD thesis, MIT Dept. of Aero. and Astro., 1998.
- [15]O. Enoksson. Shape optimization in compressible inviscid flow. Licentiate Thesis LiU-TEK-LIC-2000:31, 2000. Institute of Technology, Lindköpings University, Dept. of Math.
- [16]J. Reuther et al. Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers, part 1. *J. Aircraft*, 36(1):51–60, 1999.
- [17]P.D. Frank and G.R. Shubin. A comparison of optimization-based approaches for a model computational aerodynamics design problem. *Journal of Computational Physics*, 98:74–89, 1992.
- [18]M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Control*, 65:393–415, 2000.
- [19]B. E. Green and J. L. Whitesides. A method for the constrained design of natural laminar flow airfoils. *AIAA Paper*, (96-2502), 1996.
- [20]A. Hanifi, D. S. Henningson, S. Hein, and M. Bertolotti, F. P. and Simen. Linear non-local instability analysis - the linear NOLOT code. *FFA TN*, 1994-54, 1994.
- [21]T. Herbert. Parabolized stability equations. *Annu. Rev. Fluid Mech.*, 29:245–283, 1997.
- [22]A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.
- [23]A. Jameson, N.A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier-Stokes equations. *AIAA Paper no 97-0101*, 1997.
- [24]R.D. Joslin. Overview of laminar flow control. Technical Report 1998-208705, NASA, Langley Research Center, Hampton, Virginia, October 1998.
- [25]J.-M. Lee, D. P. Schrage, and D. N. Mavris. Development of subsonic transports with laminar flow wings. *AIAA Paper*, (98-0406), 1998.
- [26]M. J. Lighthill. A new method of two-dimensional aerodynamic design. Technical report, ARC, 1945. Rand M 2112.
- [27]J.L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations*. Springer-Verlag, New York, 1971. Translated by S.K. Mitter.

- [28]M. R. Malik. Prediction and control of transition in supersonic and hypersonic boundary layers. *AIAA J.*, 27:1487–1493, 1989.
- [29]M. R. Malik and P. Balakumar. Nonparallel stability of rotating disk flow using PSE. In M.Y. Hussaini, A. Kumar, and C.L. Streett, editors, *Instability, Transition and Turbulence*, pages 168–180. Springer, 1992.
- [30]V. M. Manning and I. M. Kroo. Multidisciplinary optimization of a natural laminar flow supersonic aircraft. *AIAA Paper*, (99-3102), 1999.
- [31]B. Mohammadi. A new optimal shape procedure for inviscid and viscous turbulent flows. *Int. J. Numer. Meth. Fluids*, 25:183–203, 1997.
- [32]J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research, 1999.
- [33]O. Pironneau. On optimal profiles in Stokes flow. *J. Fluid Mech.*, 59:117–128, 1973.
- [34]J. O. Pralits. Towards optimal design of vehicles with low drag: Applications to sensitivity analysis and optimal control. Licentiate thesis KTH/MEK/TR-01/07, Royal Institute of Technology, SE-172 90 Stockholm, Sweden, 2001.
- [35]J. O. Pralits, C. Airiau, A. Hanifi, and D. S. Henningson. Sensitivity analysis using adjoint parabolized stability equations for compressible flows. *Flow, Turbulence and Combustion*, 65:321–346, 2000.
- [36]J. O. Pralits and A. Hanifi. Optimization of steady suction for disturbance control on infinite swept wings. *Phys. Fluids*, 15(9), 2003.
- [37]J. O. Pralits, A. Hanifi, and D. S. Henningson. Adjoint-based optimization of steady suction for disturbance control in incompressible flows. *J. Fluid Mech.*, 467:129–161, 2002.
- [38]G. B. Schubauer and H. K. Skramstad. Laminar-boundary-layer oscillations and transition on a flat plate. *NACA Tech. Note No. 909*, 1948.
- [39]M. Simen. Local and non-local stability theory of spatially varying flows. In M. Hussaini, A. Kumar, and C. Streett, editors, *Instability, Transition and Turbulence*, pages 181–201. Springer, 1992.
- [40]A. M. O. Smith and N. Gamberoni. Transition, pressure gradient and stability theory. Technical Report ES 26388, Douglas Aircraft Co., 1956.
- [41]B.I. Soemarwoto. *Multi-Point Aerodynamic Design by Optimization*. PhD thesis, Delft University of Technology, Faculty of Aerospace Engineering, P.O. Box 5058, 2600 GB Delft, Netherlands, 1996.

- [42]C. Sung and J.H. Kwon. Accurate aerodynamic sensitivity analysis using adjoint equations. *AIAA Journal*, 38(2):243–250, 2000.
- [43]J. L. van Ingen. A suggested semiempirical method for the calculation of the boundary layer transition region. Technical Report VTH-74, Department of Aeronautical Engineering, University of Delft, 1956.
- [44]S. Wallin and A. V. Johansson. An explicit algebraic Reynolds stress model for incompressible and compressible turbulent flows. *J. Fluid Mech.*, 403:89–132, 2000.
- [45]C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. L-BFGS-B: Fortran subroutines for large scale bound constrained optimization. Technical Report NAM-11, EECS Department, Northwestern University, 1994.
- [46]Y. H. Zurigat, A. H. Nayfeh, and J. A. Masad. Effect of pressure gradient on the stability of compressible boundary layers. *AIAA Paper*, (90-1451), 1990. 21st Fluid Dynamics, Plasma Dynamics and Lasers Conference, Seattle, WA.

Recent licentiate theses from the Department of Information Technology

- 2002-008** Henrik Lundgren: *Implementation and Real-world Evaluation of Routing Protocols for Wireless Ad hoc Networks*
- 2003-001** Per Sundqvist: *Preconditioners and Fundamental Solutions*
- 2003-002** Jenny Persson: *Basic Values in Software Development and Organizational Change*
- 2003-003** Inger Boivie: *Usability and Users' Health Issues in Systems Development*
- 2003-004** Malin Ljungberg: *Handling of Curvilinear Coordinates in a PDE Solver Framework*
- 2003-005** Mats Ekman: *Urban Water Management - Modelling, Simulation and Control of the Activated Sludge Process*
- 2003-006** Tomas Olsson: *Bootstrapping and Decentralizing Recommender Systems*
- 2003-007** Maria Karlsson: *Market Based Programming and Resource Allocation*
- 2003-008** Zoran Radovic: *Efficient Synchronization and Coherence for Nonuniform Communication Architectures*
- 2003-009** Martin Karlsson: *Cache Memory Design Trade-offs for Current and Emerging Workloads*
- 2003-010** Dan Wallin: *Exploiting Data Locality in Adaptive Architectures*
- 2003-011** Tobias Amnell: *Code Synthesis for Timed Automata*
- 2003-012** Olivier Amoignon: *Adjoint-Based Aerodynamic Shape Optimization*



UPPSALA
UNIVERSITET