Final Report

for

NASA Ames Cooperative Agreement
NCC2-5049

July 1996

Department of Mechanical and Aeronautical Engineering
University of California, Davis

Optimum Shape Design using Automatic Differentiation in Reverse Mode [*]

M. Hafez, B. Mohammadi and O. Pironneau
U. C. DAVIS and INRIA

Summary

This paper is devoted to show how to use automatic differentiation in reverse mode as a powerful tool in optimization procedures. It is also shown that for aerodynamic applications the gradients have to be as accurate as possible. In particular, the effect of having the exact gradient of the first or second order spatial discretization schemes is presented. We show that the loss of precision in the gradient affects not only the convergence, but also the final shape. Both two and three dimensional configurations of transonic and supersonic flows have been investigated. These cases involve up to several thousand of control parameters.

1 Introduction

The problems of interest here belong to optimal shape design in aeronautics. Some implicit cost functional has to be minimized over a set of possible states, under the constraint that the state equations (steady Euler) are satisfied.

When using gradient based methods for optimization, we need the gradient of the cost function with respect to control parameter variations. This is a severe limitation if a direct method based on successive cost function evaluation is used. Indeed, the cost of one evaluation of the gradient will be proportional to the number of control parameters.

We use the Odyssée system [1,2] in reverse mode to produce the Jacobian of the cost function with respect to the control variables. Odyssée takes as input a FORTRAN 77 program and a set of variables and returns a new FORTRAN 77 program computing the derivatives of the original function with respect to the given variables. This gradient has been used in a projected conjugate gradient method for minimization.

As mentioned before, in optimization procedures we have to compute the effects of control variables variations on the cost J (i.e. dJ/dx_c). Therefore, if a control point is moved, this variation has to be propagated to all the mesh nodes. Because the target geometries are usually described by an unstructured mesh, we have developped a corresponding framework for shape deformations to take into account these deformations over the meshes. The meshes used here contain only triangles in 2D and tetraedra in 3D.

2 Control Problem Formulation

In this paper the aim is to minimize a functional $J(x, U(x))$ under geometric and aerodynamic constraints. Here, x indicates the geometrical description of a configuration and the flow pattern around this shape is the solution of the steady Euler system of fluid dynamics in conservation form: $\nabla.(F(U)) = 0$, where U is the vector of conservative variables (i.e. $U = (\rho, \rho \vec{u}, \rho(C_v T + \frac{1}{2}|u|^2))^t)$, F represents the advective operator. This system has 4 equations in 2D (5 equations in 3D) for 5 variables (6 variables in 3D) and the system is closed using the equation of state $p = p(\rho, T)$.

2.1 The Reverse Mode

To calculate the gradient of J under aerodynamic constraints, automatic differentiation in reverse mode has been used. In this approach, the lines of the programs describing the relations between the variation of the design variables and the cost function including the grid and the 'steady' flow equations are multiplied by parameters ($\bar{p}$) and an augmented Lagrangian (L) is constructed. The values of the parameters are obtained from the optimality conditions (i.e. that the first variations of L with respect to intermediate variables vanish). The

solution can be always obtained simply by back substitution (hence the notion of reverse mode). Once the parameters are evaluated, the gradient of L can be easily calculated (a simple example is given in the appendix).

In this approach no adjoint system is solved and the work in the back substitution step is equivalent to one explicit iteration of the governing equations [9].

The advantages over the other methods are clear. In particular, this leads to the exact gradient of the discrete cost function with respect to the control variables and the computational time for the gradient is independant of the number of controls.

This gradient is then used in a conjugate gradient method to solve the optimization problem. In [4,5,6], the gradients obtained by **Odyssée** have been compared with those obtained using finite differences for similar problems.

2.2 The Conjugate Gradient Method

Our minimization tool is quite simple. It is based on a conjugate gradient method with optimal descent step. We use projection to take into account the local geometrical constraints and global constraints are taken into account in the cost function using Lagrange multipliers.

More precisely, the algorithm is as follows (we denote $J(x, U(x))$ by $J(x)$:

$$x_0 given,$$

do 1 iteration of the steepest descent algorithm,

$$for \quad n = 2, 3, ... \quad do$$

$$\gamma = \frac{|\nabla_x J(x^{n-1})|^2}{|\nabla_x J(x^{n-2})|^2},$$

$$h^n = -\nabla_x J(x^{n-1}) + \gamma h^{n-1},$$

$$x^n = x^{n-1} + \lambda^n h^n,$$

where,

$$\lambda^n = min_\lambda J(\tilde{x}, U(\tilde{x})).$$

with $\tilde{x} = x^{n-1} - \lambda \nabla_x J(x^{n-1})$ and

$$\nabla_x J = \frac{\partial J}{\partial x} + (\frac{\partial J}{\partial U})^T (\frac{\partial U}{\partial x}).$$

This algorithm converges to a local minimum of J.

2.3 Flow Solver

The NSC2KE solver uses a finite element/finite volume formulation on unstructured meshes involving triangles in 2D and tetraedra in 3D. Second order accuracy in space has been achieved using a MUSCL type reconstruction and limiters have been used to prevent oscillations. The time dependant equation $(\partial U/\partial t + \nabla.F(U) = 0)$ is marched in time to a steady state. The time discretization is based on a 4-stage Runge-Kutta scheme. We will show that for the optimization procedure it is important to have a gradient including all of these ingredients (especially, the second order MUSCL reconstruction step).

2.4 Geometry Modifications and Control Points Definition

In 2D, the control points are fitted by a cubic spline. The splines have two features. They have a smoothing effect on the variations of the control points (δx_c) and they propagate these variations to the other body points which are not control points (δx_w).

For 3D applications, the use of generalized surface splines is quite complicated and involves CAD concepts and deriving these objects is more difficult than the fluid solver. The present unstructured framework [3] for geometry modifications is based on the following:

1. All the nodes on the shape are control points.

2. To avoid oscillations, a smoothing operator is defined over the shape. This can be, for instance, a few Jacobi iterations to solve $((I - \epsilon\Delta)\delta\tilde{x}_w = \delta x_w)$, where $\delta\tilde{x}_w$ is the smoothed shape variation for the shape nodes and δx_w is the variation given by the optimization tool. Once (x_w) known, we have to expand these variations overall the mesh. This is done by solving an elliptic system. These tools have also been derived by the automatic differentiation procedure.

2.5 Geometrical Constraints

The present geometrical constraints are of two types. The first one is imposed by defining two limiting surfaces (curves in 2D) between which shape variations are allowed. As all shapes in this paper are of wing type, the second constraint is that the original planform should remain unchanged. This means for instance in 2D that the leading and trailing edge are frozen.

2

3 Results

Two and three dimensional results of inverse problems and drag reduction for airfoils and wings are presented for transonic and supersonic flows.

3.1 An Inverse Problem in 2D

The first case consists of an inverse problem based on a given pressure distribution. The cost function is given by $J(x) = \frac{1}{2}\int_x |p_x - p_{target}|^2 dx$, where, p_{target} is a given target pressure and p_x the actual flow pressure. The design takes place at Mach number 0.85 and zero angle of incidence. The initial shape is the RAE2822. The target pressure corresponds to the same airfoil deformed by about 30 percent on the upper surface and 20 percent on the lower surface. This leads to a shift of the shocks to the right by about 20 percent of the chord. There are 20 control points on the airfoil and 60 total nodes. This is a quite coarse mesh and enables us to compare the Odyssée gradients with finite differences (see Fig. 1).

The cost function has been computed using a second order scheme but the gradient has been computed either by a first or a second order scheme. This is to show that this loss of precision for the gradient impacts both the convergence and the final shape.

3.2 A drag reduction problem in 2D

The aim here is to reduce the shock-induced drag for a RAE2822 profile at Mach number 0.8 and zero angle of incidence. The shock on the upper surface is quite close to the trailing edge and is more difficult to remove as in this region geometrical constraints are more important. On the other hand, the shock on the lower surface is easier to remove.

As in [7,8], consider as cost function $J(x) = \frac{1}{2}\int_\Gamma |p - p_i|^2 dx + 10C_d$, where p_i is the original pressure distribution and C_d the drag coefficient. The first term forces the profile to remain as much as possible close to the original shape. Moreover, the shape variation is limited to 5 percent. The drag (resp. lift) has been reduced from $2.1 10^{-2}$ (resp. 0.292) to $1.25 10^{-2}$ (resp. 0.291) with the second order gradient and to $1.53 10^{-2}$ (resp. 0.289) using the first order one. The airfoil volume's has almost remained unchanged.

3.3 An Inverse Problem in 3D

We consider a pressure recovery problem in 3D. The original shape here is the ONERA M6 wing. The target shape is the same wing deformed on its upper surface by 30 percent. Our aim is to recover this shape using the corresponding pressure distribution starting from the M6 wing. The section definitions of the wing are not used. Also, the cross sections have been computed by interpolation (see Fig. 6). All the wall mesh points are control points. In this case, we have about 700 nodes on the wing. It is therefore impossible to treat this case without the adjoint (inverse) mode. The important remark here is that when we use only first order gradient, the target shape is not correctly recovered.

3.4 Wave-Drag Reduction in 3D

The aim here is to reduce the shock-induced drag over an M6 type wing. The original wing is an M6 wing with the upper surface deformed by 10 percent to obtain a nonsymmetric wing. The mesh has around 10^5 tetraedra and there are about 2000 control points on the wing (see. Fig. 11). The design take place at Mach number 0.84 and incidence of 3.06. This configuration involves a λ-shock on the upper surface and our aim is to produce a wing as close as possible to the original with smoothed shocks. We will use the second order gradient in this computation. The cost function is $J(x) = \frac{1}{2}\int_\Gamma |p - p_0|^2 dx + 10C_d + |C_l - C_l^0|$, where C_l and C_l^0 are the actual and initial lift coefficients. The drag has been reduced by about 10 percent.

3.5 A supersonic case

This is an inverse problem at Mach 3 over an Naca0014 airfoil. The aim is to show that the extension of our approach to supersonic flows does not introduce any particular difficulty even at boundaries.

4 Concluding Remarks

A new approach involving the reverse mode of automatic differentiation has been presented. A general framework for treating geometries with unstructured discretization has been introduced for both two and three dimensional configurations. Preliminary examples show the ability of the method to treat inverse and control problems.

The inverse mode is shown to be a powerful tool for providing the exact gradient of the discrete operators. Thus, 3D cases with several thousands of control points are possible to calculate.

It has also been pointed out that the gradient should include all the ingredients of the discrete operators. The accuracy of the gradient impacts not only the convergence but also the final shapes.

It is important to notice that, except for the gradient which use a backward time integration procedure, all the calculations have been done by explicit schemes. A parallel implementation of this approach is therefore quite simple.

Future works will include more extensive validation of these techniques in 3D configurations.

5   References

[1] J.C. Gilbert, G. Le Vey, J. Masse,*La différentiation automatique de fonctions représentées par des programmes*, Rapport de Recherche INRIA 1557, (1991).

[2] N. Rostaing-Schmidt, *Différentiation automatique: Application à un problème d'optimisation en météorologie*, Ph.D. Thesis, University of Nice, (1993).

[3] B. Mohammadi,*Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence*, submitted to 35th AIAA meeting, Reno, 1997.

[4] J.M. Malé, B. Mohammadi, N. Rostaing-Schmidt,*Direct and Reverse Modes of Automatic Differentiation of Programs for Inverse Problems: Application to Optimum Shapes Design* SIAM workshop on computational differentiation, Santafe (1996).

[5] B. Mohammadi, O. Pironneau,*New Progress in Optimum Shapes Design*, CFD Review (1995), Hafez-Oshima Eds.

[6] B. Mohammadi, O. Pironneau, *Automatic Differentiation and Nonlinear PDE*, ECCOMAS 96, Paris.

[7] A. Jameson,*Optimum Aerodynamic Design via Boundary Control*, AGARD Report 803, Von Karman Institute Courses (1994).

[8] O. Pironneau,*Optimal Shape Design for Elliptic Systems*, Springer-Verlag, New York (1984).

6   Appendix

We give a simple example of the automatic differentiation in reverse mode. Consider a cost function f

and the design variables x such that $f(u(x))$, where $u(x)$ is the solution of the flow equations. Assume the following FORTRAN 77program:

```
u1=x
u2=x**2+2*u1
f=u1+u2
```

In automatic differentiation in reverse mode, we consider the lines of the program as constraints and associate to each of them a Lagrange multiplier and define an augmented Lagrangian as follows:

$$L = u_1 + u_2 + p_1(u_2 - x^2 - 2u_1) + p_2(u_1 - x).$$

We know that at the solution we have:

$$\frac{\partial L}{\partial u_1} = 1 - 2p_1 + p_2 = 0,$$

$$\frac{\partial L}{\partial u_2} = 1 + p_1 = 0.$$

We notice that to find p_i, we have to solve the previous set of equations in "reverse" order. Once p_i known, we have:

$$\frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} = -2p_1x - p_2,$$
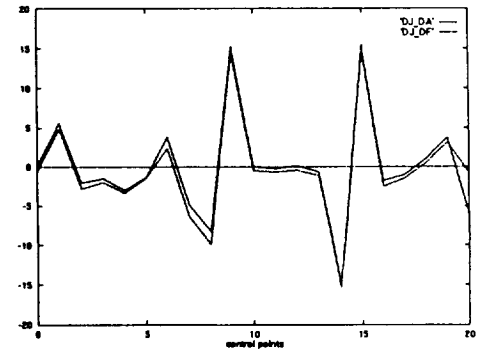
which is the Jacobian of f.

Figure 1: *Inverse problem with inverse mode: comparison of the gradients obtained by Odyssée and finite differences.*
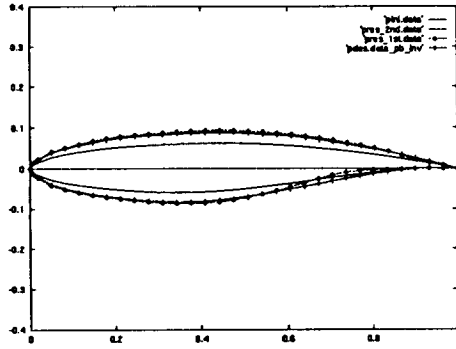
4

Figure 2: *Inverse problem with inverse mode: initial, target and computed shapes using the first and second order Jacobians.*
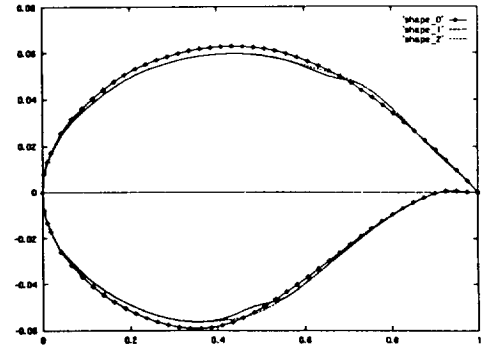

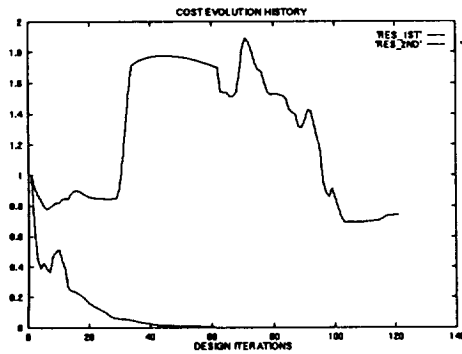Figure 5: *Drag reduction: initial and final shapes obtained with the first and second order gradients.*

Figure 3: *Inverse problem with inverse mode: convergence histories for the cost using first and second order gradients.*
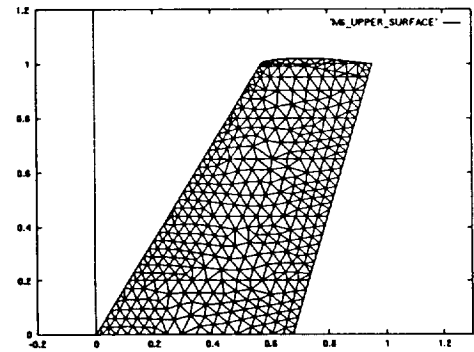

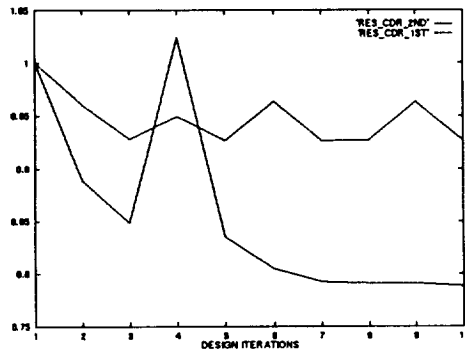Figure 6: *3D inverse problem: M6 wing, upper surface discretization.*

Figure 4: *Drag reduction: convergence histories for the cost function using second order operator and first or second order gradients.*
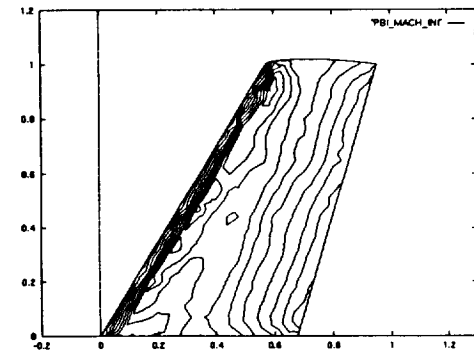
Figure 7: *3D inverse problem: Iso-Mach contours over the M6 wing's upper surface (in the range [0.3,1.4] with 40 contours).*

5
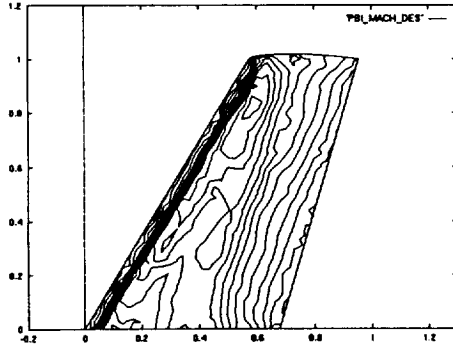
Figure 8: *3D inverse problem: Iso-Mach contours over the target wing's upper surface (in the range [0.3,1.4] with 40 contours).*
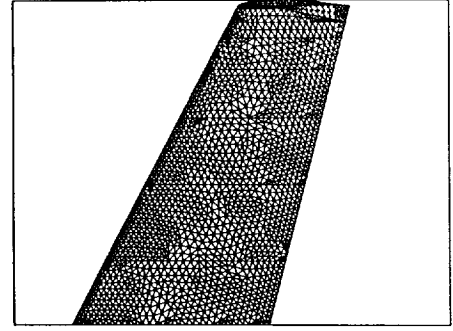
Figure 11: *3D Drag reduction: nonsymmetric wing, upper surface discretization, all these points are control points.*
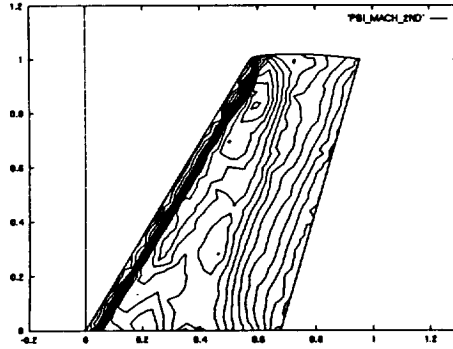
Figure 9: *3D inverse problem: Iso-Mach contours over the upper surface of the shape obtained using the second order gradient (uniform discretization of the interval [0.3,1.4] with 40 isovalues).*
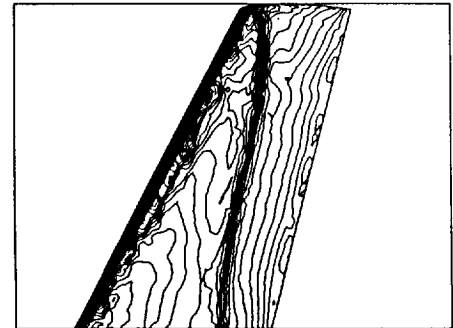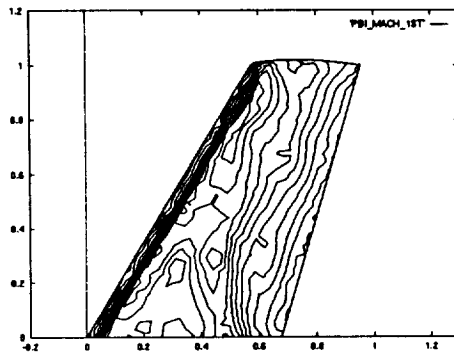

Figure 12: *3D drag reduction: Iso-Mach contours over the initial wing's upper surface.*

Figure 10: *3D inverse problem: Iso-Mach contours over the upper surface of the shape obtained using the first order gradient (in the range [0.3,1.4] with 40 contours).*
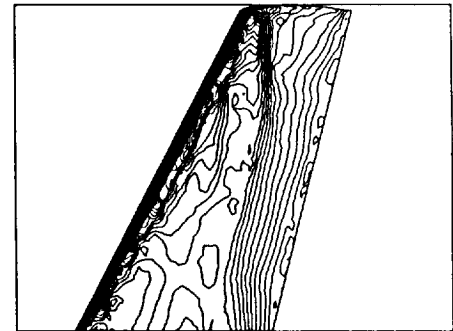
Figure 13: *3D drag reduction: Iso-Mach contours over the upper surface of the shape obtained using the second order gradient.*
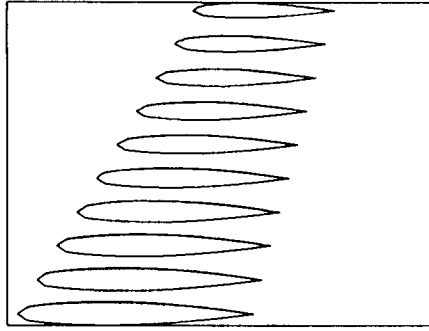
6

Figure 14: *3D drag reduction: shape cross-section for the initial and the optimized shape after 5 design iterations.*
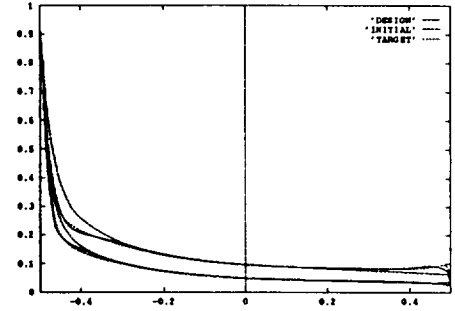

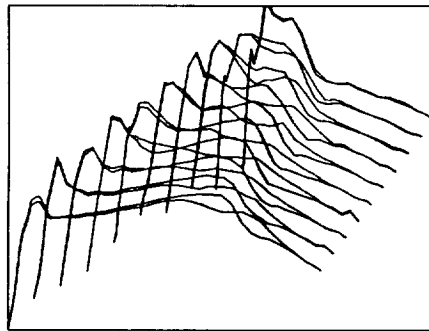Figure 17: *Supersonic flow at M=3. Pressure distribution: initial, target and computed.*

Figure 15: *3D drag reduction: pressure cross-section over the initial and the optimized shape after 5 design iterations.*
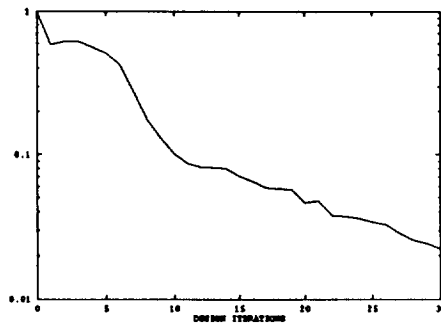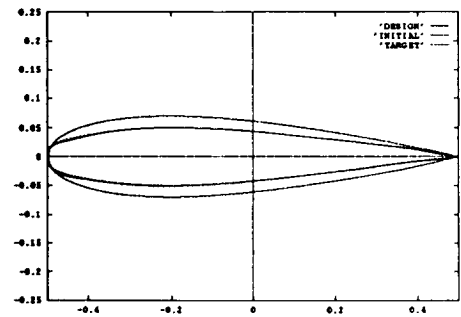

Figure 18: *Supersonic flow at M=3. Initial, target and final shapes.*

Figure 16: *Supersonic flow at M=3. Convergence history.*

7