AIAA JOURNAL
Vol. 34, No. 9, September 1996

# Pseudotime Method for Shape Design of Euler Flows

Angelo Iollo*
*Politecnico di Torino, 10129 Torino, Italy*
Geojoe Kuruvila†
*ViGYAN, Inc., Hampton, Virginia 23681-0001*
and
Shlomo Ta'asan‡
*Institute for Computer Applications in Science and Engineering, Hampton, Virginia 23681-0001*

**We exploit a novel idea for the optimization of flows governed by the Euler equations. The algorithm consists of marching on the design hypersurface while improving the distance to the state and costate hypersurfaces. We consider the problem of matching the pressure distribution to a desired one, subject to the Euler equations, for both subsonic and supersonic flows. We limited our investigation to two-dimensional test cases. The rate of convergence to the minimum for the cases considered is three to four times slower than that of the analysis problem. Results are given for Ringleb flow and a shockless compression case.**

## I. Introduction

In recent years there has been a renewed interest in optimal design in fluid mechanics. Faster computers and reliable numerical simulations make feasible some of the aerodynamics optimal design problems that are of engineering interest.

The preceding statement becomes only partially true when we consider either flows governed by Euler or Navier–Stokes equations or complicated geometrical configurations with many design variables. For such cases, shape optimization seems to be still not practical because of extremely time-consuming computation. The present work aims at a flexible and feasible approach for such intensive computational problems by applying a novel algorithm proposed by Ta'asan.[1]

The problem of finding a shape that achieves given performance has been attacked by means of inverse problem formulations.[2–5] These methods have in common the advantage of being solved at the same cost of an analysis problem. They are in general not extendable to three dimensions. Moreover the set of problems that can be solved by means of inverse design is limited.

A more general framework is to consider aerodynamics design problems as optimization problems. From the mathematical viewpoint the problem is to find $U$ such that

$$\begin{cases} U \in \mathcal{U} \\ \mathcal{E}(U) \leq \mathcal{E}(V) \qquad \forall V \in \mathcal{U} \end{cases}$$

where $\mathcal{U}$ is a given set and $\mathcal{E}$ is a real-valued functional defined on $\mathcal{U}$.

Shape design optimization problems are tightly related to control of a system governed by partial differential equations where the controls (the design variables) are on the boundary. Lions[6] set the mathematical framework for such problems. The theory is concerned mainly with linear systems and is devoted "(i) to obtain necessary (or possibly necessary and sufficient) conditions for $U$ to be an extremum (or minimum), (ii) to study the structure and properties of the equations expressing these conditions, (iii) to obtain constructive algorithms amenable to numerical computations for the approximation of $U$."

Pironneau[7,8] applied control theory to the minimum-drag problem in Stokes flows and subsequently in flows governed by the incompressible Navier–Stokes equations. Because a Navier–Stokes solver was not available, some solutions were obtained using simpler models.[9]

For the Euler equations, Jameson[10] proposed an optimization algorithm based on the adjoint method to compute the gradient of the functional with respect to the design variables. Conformal mapping was used to control the shape of the wing to be designed. Iollo et al.[11] studied the case of Euler flows with embedded shocks for a one-dimensional case and discussed the boundary conditions for the adjoint equations. At the shock location it was shown that further conditions are needed for the adjoint equation to be well posed. Subsequently, Iollo and Salas[12] extended these results to two-dimensional flows and presented computations with higher-order spatial accuracy.[12]

The high computational cost for solving optimization problems governed by fluid dynamics equations comes from several sources. The first is the cost of a single analysis, which for the Navier–Stokes equations in three dimensions is on the order of a few Cray hours. Another source is the fact that a repeated solution of the flow equations may be required for gradient methods. An additional significant cost may arise from the calculation of the gradient of the functional.

The use of adjoint methods eliminates the unnecessary cost resulting from computation of the gradient and is much more efficient compared with other methods including finite differences and sensitivity analysis. It requires the computation of an extra system of partial differential equations (PDEs), namely, the costate equations, but the total cost for gradient calculation is independent of the number of design variables. A comparison study of calculating gradients using adjoint methods and finite difference methods was done by Beux and Dervieux.[13] They also solved pressure reconstruction problems for compressible internal flows, comparing the performances of several algorithms. Flows with embedded shocks were not considered in this work.

The adjoint method, being an efficient method for calculating the gradient, does not address the computational expense related to the number of gradient iterations required to reach the minimum. In general, the number of iterations required to achieve the minimum grows more than linearly with the number of design variables used, making infeasible design problems in three dimensions with many design variables.

In another contest Rizk[14] introduced the idea of minimizing a functional by simultaneously updating flow variables and design variables. With this technique he was able to reduce the cost of optimization to that of $L$ flow analysis, where $L$ is the number of design variables. Independently, Ta'asan[15] proposed an algorithm to reduce

Received August 15, 1995; revision received May 14, 1996; accepted for publication May 17, 1996. 

*Postdoctoral Researcher, Dipartimento di Ingegneria Aeronautica e Spaziale, Corso Duca degli Abruzzi 24.

†Research Engineer; currently Principal Engineer, Advanced Transport Aircraft Development, McDonnell Douglas Aerospace, Long Beach, CA 90810-1870.

‡Senior Scientist, NASA Langley Research Center; currently Associate Professor, Department of Mathematics, Carnegie–Mellon University, Pittsburgh, PA 15213.

the cost of the optimization to that of a single analysis, namely, the one shot method. The idea is to solve the flow equations, the costate equations, and the optimality condition at the same time. The main idea in that algorithm was to perform the optimization iteration on coarse grids that are used anyway in the multigrid process. Small numbers of design variables were considered in that case.

Ta'asan et al.[16] applied this technique to a potential flow and extended the method to cases of moderate numbers of design variables. Different design variables are associated with different grids depending on the smoothness of the shape functions associated with them and are updated on these grids. The performance of this algorithm was practically independent of the number of design variables.

Arian and Ta'asan[17] extended the one shot method to infinite-dimensional design space. The main idea of the method was to construct a relaxation that smoothes the errors in the design variables. Application to control problems and shape design problems have demonstrated solution of the full optimization problem in a cost comparable to that of solving the analysis just a few times, independent of the number of design variables (experiments using up to 128 design variables have been done).

Beux and Dervieux[18] proposed a hierarchical strategy in which the number of design variables is progressively increased, performing a multilevel optimization that seems to render the computational cost independent from the number of design variables.

The drawbacks of the one shot methods are their programming complexity and the fact that their use is limited to multigrid solvers. This was the motivation for the study of a new type of solution strategy for optimization problems governed by PDEs.[1] The goal was to try to get methods that solve the optimization problem in a cost comparable to that of the analysis. The emphasize was on simplicity and flexibility to work in existing frameworks that do not necessarily involve multigrid methods.

The main observation is the following. The solution of the optimization problem lies on the intersection of the sets that contain state and costate equations solutions. Gradient-based methods (including adjoint formulations) can be viewed as marching along the intersection of these sets. This is an expensive process because each step requires the solution of two PDEs, i.e., the Euler equations and the costate equations. The idea of the pseudotime method was to perform the marching satisfying only the design equation and the boundary conditions for state and costate equations, while iterating to solve both state and costate equations. The cost of such an iteration per step is significantly smaller than that of gradient-based methods. Its convergence has been shown by Ta'asan[1] to be independent of the number of design variables.

In the present paper we extend the application of the pseudotime method to optimization problems involving the Euler equations. In fact, for such equations a simple application of the pseudotime algorithm is not possible, since the design equation viewed as an equation for the design variables may be singular.

Using the pseudotime method the cost of optimization becomes of the same order as that of analysis. Moreover, we present an algorithm that may be implemented with no substantial changes to existing adjoint-based optimization codes. Numerical evidence indicates that the method converges at a rate that is independent of the number of design variables.

## II. Problem Statement

The Euler equations in two dimensions are given by

$$\boldsymbol{U}_t + \boldsymbol{F}_x + \boldsymbol{G}_y = 0 \tag{1}$$

where

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{bmatrix} \qquad \boldsymbol{F} = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ u(\rho e + p) \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ v(\rho e + p) \end{bmatrix}$$

with

$a$ = speed of sound
$e$ = specific total energy
$p$ = pressure

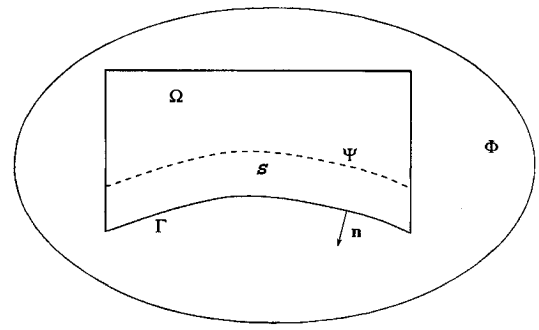


**Fig. 1 Model problem: $\Phi$ is the domain of definition of the Euler equations; $\Omega$ is an included subdomain with boundary $\Gamma$ and normal to the boundary $n$; and $S$ is the small stripe in the vicinity of the boundary in which the optimization problem is solved at steps 4 and 5 of the pseudotime algorithm.**

$u = x$ component of the velocity vector
$v = y$ component of the velocity vector
$\gamma$ = ratio of specific heats
$\kappa = (\gamma - 1)/2$
$\rho$ = density
and $p = \kappa\rho(2e - u^2 - v^2)$.

Furthermore let

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{U}} = \boldsymbol{A}(\boldsymbol{U}) \qquad \text{and} \qquad \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{U}} = \boldsymbol{B}(\boldsymbol{U}) \tag{2}$$

We assume that these equations are defined on a domain $\Phi$ that includes a subdomain $\Omega$ whose boundary is denoted by $\Gamma$. On the boundary we define a curvilinear coordinate $s$ and a normal $\boldsymbol{n} = (n_x, n_y)$ pointing outward. A real-valued functional $\mathcal{E}[\Gamma, V(\Gamma)]$ is given, where $V(\Gamma)$ is the solution of the Euler equations with boundary conditions on $\Gamma$. The optimization problem that we study is as follows: minimize the functional $\mathcal{E}[\Gamma, V(\Gamma)]$ over all of the admissible shapes of the boundary $\Gamma$.

We focus on the following model problem. The subdomain $\Omega$ is represented by a nozzle; see Fig. 1. At the inlet, total pressure, total temperature, and the ratio $\sigma = v/u$ are fixed. At the outlet, if the flow is subsonic, the static pressure is fixed, and at the solid walls the impermeability condition $un_x + vn_y = 0$ is enforced. The upper wall is kept fixed. The lower wall $\Theta$ is represented by mean of the parameterization

$$y(\Theta) = \sum_i \alpha_i f_i(x) \tag{3}$$

where the functions $f_i(x)$ are some shape functions and $\alpha = (\alpha_1, \ldots, \alpha_i, \ldots)$ is the corresponding set of shape coefficients. Given a desirable lower wall pressure distribution $p^*(x)$ and denoting by $p^\Theta(x)$ the actual one on the lower wall, the optimization problem consists in finding a set of shape coefficients $\alpha_i$ such that the functional

$$\mathcal{E} = \frac{1}{2}\int_a^b (p^\Theta - p^*)^2 \, dx \tag{4}$$

is minimized.

## III. Optimality Conditions

The optimality conditions are derived by introducing Lagrange multipliers and considering the augmented functional

$$\mathcal{L}(\boldsymbol{U}, \alpha, \boldsymbol{\Lambda}, \mu) = \mathcal{E}(\alpha, \boldsymbol{U}) + \int_\Omega {}^t\boldsymbol{\Lambda}(\boldsymbol{A}\boldsymbol{U}_x + \boldsymbol{B}\boldsymbol{U}_y)\, d\Omega + \int_\Theta \mu\rho \boldsymbol{V}\cdot\boldsymbol{n}\, ds \tag{5}$$

where $\boldsymbol{V} = (u, v)$. The vector $\boldsymbol{\Lambda}(x, y) = {}^t(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$, and the scalar $\mu(s)$ are the Lagrange multipliers.

Calculating the variation of the functional $\mathcal{L}$ with respect to the variation of the functions $\boldsymbol{U}$, $\boldsymbol{\Lambda}$, and $\mu$ and the parameters $\alpha_i$, respectively, we obtain[12]

$$\delta\mathcal{L}_U = \int_a^b \left.\frac{\partial p}{\partial \boldsymbol{U}}\right|_\Theta (p^\Theta - p^*)\tilde{\boldsymbol{U}}\,dx + \int_\Gamma {}^t\boldsymbol{\Lambda}(\boldsymbol{A}n_x + \boldsymbol{B}n_y)\tilde{\boldsymbol{U}}\,ds - \int_\Omega ({}^t\boldsymbol{\Lambda}_x\boldsymbol{A} + {}^t\boldsymbol{\Lambda}_y\boldsymbol{B})\tilde{\boldsymbol{U}}\,d\Omega + \int_\Theta \mu\boldsymbol{n}\frac{\partial \rho\boldsymbol{V}}{\partial \boldsymbol{U}}\tilde{\boldsymbol{U}}\,ds \tag{6}$$

where

$$\frac{\partial p}{\partial \boldsymbol{U}} = 2k\left(\frac{u^2+v^2}{2}, -u, -v, 1\right) \quad \text{and} \quad \frac{\partial \rho\boldsymbol{V}}{\partial \boldsymbol{U}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

and

$$\delta\mathcal{L}_\Lambda = \int_\Omega {}^t\tilde{\boldsymbol{\Lambda}}(\boldsymbol{A}\boldsymbol{U}_x + \boldsymbol{B}\boldsymbol{U}_y)\,d\Omega \tag{7}$$

$$\delta\mathcal{L}_\mu = \int_\Theta \tilde{\mu}\rho\boldsymbol{V}\cdot\boldsymbol{n}\,ds \tag{8}$$

$$\delta\mathcal{L}_\alpha = \sum_i \left[\int_a^b \left.\frac{dp}{dy}\right|_\Theta (p^\Theta - p^*)\,f_i\,dx + \int_\Theta {}^t\boldsymbol{\Lambda}(\boldsymbol{A}\boldsymbol{U}_x + \boldsymbol{B}\boldsymbol{U}_y)\,f_i\,\cos\theta\,ds + \int_\Theta \mu\frac{\partial(\rho\boldsymbol{V})}{\partial y}\cdot\boldsymbol{n}\,f_i\,ds - \int_\Theta \mu\rho\boldsymbol{V}\cdot\boldsymbol{t}\frac{df_i}{dx}\cos^2\theta\,ds + \int_\Theta \mu\rho\boldsymbol{V}\cdot\boldsymbol{n}\frac{df_i}{dx}\sin\theta\cos\theta\,ds\right]\tilde{\alpha}_i \tag{9}$$

where $\theta$ is the angle between the normal $\boldsymbol{n}$ and the $y$ axis, $\boldsymbol{t} = (-n_y, n_x)$, and $\tilde{\boldsymbol{U}}$, $\tilde{\boldsymbol{\Lambda}}$, $\tilde{\mu}$, and $\tilde{\alpha}_i$ are the variations of the corresponding arguments.

At the minimum of the functional, for all of the possible choices of the functions $\tilde{\boldsymbol{U}}$, $\tilde{\boldsymbol{\Lambda}}$, and $\tilde{\mu}$ and of the parameters $\tilde{\alpha}$, we must have

$$\delta\mathcal{L}_U = \delta\mathcal{L}_\Lambda = \delta\mathcal{L}_\mu = \delta\mathcal{L}_\alpha = 0 \tag{10}$$

Therefore, we have

$$\delta\mathcal{L}_\Lambda = 0 \Leftrightarrow \boldsymbol{A}\boldsymbol{U}_x + \boldsymbol{B}\boldsymbol{U}_y = 0 \quad \text{on} \quad \Omega \tag{11}$$

and

$$\delta\mathcal{L}_\mu = 0 \Leftrightarrow \rho\boldsymbol{V}\cdot\boldsymbol{n} = 0 \quad \text{on} \quad \Theta$$

which are the Euler equations and boundary conditions. Furthermore,

$$\delta\mathcal{L}_U = 0 \Leftrightarrow {}^t\boldsymbol{A}\boldsymbol{\Lambda}_x + {}^t\boldsymbol{B}\boldsymbol{\Lambda}_y = 0 \quad \text{on} \quad \Omega \tag{12}$$

and

$$\left.\frac{\partial p}{\partial \boldsymbol{U}}\right|_\Theta (p^w - p^*)\cos\theta + {}^t\boldsymbol{\Lambda}(\boldsymbol{A}n_x + \boldsymbol{B}n_y) + \mu\boldsymbol{n}\frac{\partial \rho\boldsymbol{V}}{\partial \boldsymbol{U}} = 0 \quad \text{on} \quad \Theta \tag{13}$$

where

$$\mu = -[\lambda_1 + u\lambda_2 + v\lambda_3 + (\gamma e - kV^2)\lambda_4] \tag{14}$$

For the boundary condition on inlet and outlet we refer the reader to Ref. 12. Given $\boldsymbol{U}$, the set of costate equations (12–14) determine uniquely $\boldsymbol{\Lambda}$ in $\Omega$ and $\mu$ on $\Theta$. Finally, given $\alpha$ and knowing $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$, we can calculate from Eq. (9)

$$\frac{\partial\mathcal{L}}{\partial\alpha_i} = \int_a^b \left.\frac{dp}{dy}\right|_\Theta (p^\Theta - p^*)\,f_i\,dx + \int_\Theta {}^t\boldsymbol{\Lambda}(\boldsymbol{A}\boldsymbol{U}_x + \boldsymbol{B}\boldsymbol{U}_y)\,f_i\,\cos\theta\,ds + \int_\Theta \mu\frac{\partial(\rho\boldsymbol{V})}{\partial y}\cdot\boldsymbol{n}\,f_i\,ds - \int_\Theta \mu\rho\boldsymbol{V}\cdot\boldsymbol{t}\frac{df_i}{dx}\cos^2\theta\,ds + \int_\Theta \mu\rho\boldsymbol{V}\cdot\boldsymbol{n}\frac{df_i}{dx}\sin\theta\cos\theta\,ds \tag{15}$$

In case of a shock occurring in the flowfield, we split the domain of integration by means of a curve $\Upsilon$ that coincides with the shock where it exists. Then we follow the same derivation presented so far on each of the two subdomains, regarding $\Upsilon$ as a boundary.[12] The resulting extra condition for $\boldsymbol{\Lambda}$ on the shock is $\boldsymbol{\Lambda} = 0$. Note that if the shocks are not treated properly, the problem of solving the costate equations with boundary conditions is not well-posed. Jameson presented[19] results for transonic flows over airfoils where the wave drag is minimized. He does not use any special treatment for the shock, but the costate equations converge. This is because the scheme that he uses for solving the Euler equations smears the shocks over several grid points as a result of artificial viscosity.

## IV. Pseudotime Optimization Method

There are many methods for obtaining the minimum of the functional $\mathcal{L}$ knowing its gradient with respect to the design variables. Optimization algorithms based on adjoint methods involve the following steps:

1) Start with a set $\alpha_i$ of shape coefficients.

2) Enforce $\delta\mathcal{L}_\Lambda = 0$ and $\delta\mathcal{L}_\mu = 0$ by finding a $\boldsymbol{U}$ that satisfies the steady-state Euler equations and boundary conditions.

3) Enforce $\delta\mathcal{L}_U = 0$ by finding a $\boldsymbol{\Lambda}$ that satisfies the costate equations and boundary conditions.

4) Calculate $\nabla_\alpha\mathcal{L}$; if it is 0, we have found the minimum.

5) Otherwise update $\alpha$ with $\nabla_\alpha\mathcal{L}$, using a proper step size.

6) Restart from 2 until $\nabla_\alpha\mathcal{L} = 0$.

The need to repeat the preceding steps 2 and 3 many times can become prohibitively expensive for geometrically complex configurations requiring computational power near the limits of present capabilities.

Ta'asan[1] proposed an efficient way of solving the optimization problem. The main observation is the following. The solution of the optimization problem lies on the intersection of the sets that contain state and costate equations solutions. Gradient-based methods (including adjoint formulations) can be viewed as marching along the intersection of the state and costate equation solutions sets (see Fig. 2). This is an expensive process because each step requires the solution of two PDEs, i.e., the Euler equations and the costate equations. The idea of the pseudotime method was to perform the marching satisfying only the design equation and the boundary conditions for Euler and adjoint equations, while iterating to solve both state and costate equations on the entire field. Because in the case considered here the design equation is defined on

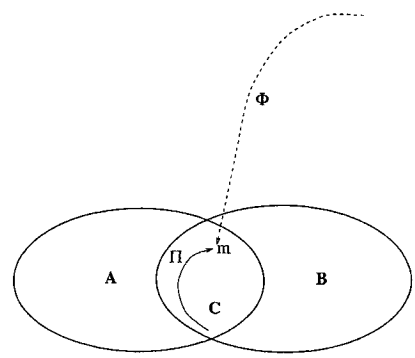


**Fig. 2 Point $m$ represents the desired optimum; set $A$ contains the points $(\alpha, \Lambda, U)$, which satisfy the Euler equations. Set $B$ contains the points $(\alpha, \Lambda, U)$, which satisfy the adjoint equations. The points in $A \cap B$ satisfy both the Euler and the adjoint equations. In the standard optimization algorithms based on adjoint method to compute the gradient, point $m$ is reached by following the path $\Pi$, which is all in $A \cap B$. At each step along $\Pi$, the state and costate equations are iterated many times. In the new approach a new path, $\Phi$, is taken. On $\Phi$ the design equation and the boundary conditions of state and costate equations are satisfied. The computational cost of working on $\Phi$ is equivalent to solving a problem one space dimension less than the original problem. The solution of the state and costate equations is achieved only when point $m$ is reached.**

the boundary, the cost of such an iteration per step is significantly smaller than that of gradient-based methods. Its convergence has been shown by Ta'asan[1] to be independent of the number of design variables.

The design equation for a wide class of problems is defined on the boundary only; thus it can be viewed as an extra boundary condition, and the design variables can be viewed as the additional variables to solve for. In some cases the design equation can be solved for the design variables, and a simple implementation of the preceding idea exists. In other cases, the design equation, viewed as an equation for the design variables keeping the state and costate fixed, may be singular, and a more involved implementation is required. This is the case for the problem considered here. In such cases it is necessary to solve for the design variables together with the state and costate variables in a small vicinity of the boundary $\mathcal{S}$.

Thus, at each step of computation on the entire field, the design equation is satisfied together with the boundary conditions for the state and costate equations. The solution on the entire field $\Omega$ affects the result of the optimization on $\mathcal{S}$ through the values of $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ on the auxiliary boundary $\Psi$; see Fig. 1.

The algorithm is as follows.

1) Start with a tentative set of $\alpha_i$.

2) March in time, on the entire field, the state equation a few steps.

3) March in time, on the entire field, the costate equation a few steps.

4) Solve in $\mathcal{S}$ the state equation with its boundary conditions and the costate equation with its boundary conditions, and compute $\nabla_\alpha \mathcal{L}$.

5) If $\nabla_\alpha \mathcal{L} = 0$, restart from step 2, repeating steps 3 and 4 until the state and costate equations are converged on the entire field. Otherwise take $\alpha_i^{n+1} = \alpha_i^n + f(\nabla_\alpha \mathcal{L})$ and go to step 4.

It is seen that steps 4 and 5 are equivalent to the first algorithm of this section, but we reduce the problem of one dimension by finding the solution of the optimization only in the stripe $\mathcal{S}$. Note that one could try to solve the problem on the boundary, in the preceding steps 4 and 5, using a direct solver, but this leads to coding complications. The way we propose here has the advantage of being a simple modification of the adjoint method and therefore can be easily implemented.

In step 5 we take $\alpha_i^{n+1} = \alpha_i^n - a\,\nabla_\alpha \mathcal{L}$, where $a$ is a parameter. This is known as the steepest descent method, and although it is very simple, it is known to be very inefficient. However, we have preferred simplicity to efficiency, since the computational cost of steps 4 and 5 is modest compared with that of steps 2 and 3.

We anticipate that the optimal number of iterations of state and costate equations to be performed on $\Omega$ (steps 2 and 3 of the preceding algorithm) was selected on the basis of experimentation. In principle, increasing the number of iterations on $\Omega$, we get closer and closer to a standard adjoint-based optimization algorithm. We found that, independently from the problem considered, the global cost of optimization was minimized by iterating 10 times state and costate equations at steps 2 and 3. An indicative breakdown of the computational costs of the preceding algorithm is given in the next section.

## V. First Optimization Experiments

We introduce a discrete grid defined as $(x_l, y_m) = [x_0 + l\Delta x, y(\Theta) + m\Delta y]$, where $\Delta x$ is constant and $\Delta y$ is a constant fraction of the local height of the nozzle.

The steady solution of the Euler equations is obtained with a time-dependent technique, in the frame of an explicit finite volume code. The conservative variables $\boldsymbol{U}$ are computed at the cell centers, and the fluxes $\boldsymbol{F}$ and $\boldsymbol{G}$ are evaluated at the cell interfaces using the approximate Riemann solver in Ref. 20. Higher-order accuracy is achieved using an essentially nonoscillatory scheme.[21] The flowfield values at cell interfaces, used as initial conditions for the Riemann problem, are reconstructed by means of a linear interpolation and using a minimod limiter. The amplitude of the integration step is chosen according to the Courant–Friedrichs–Lewy condition.

The costate equations are discretized on the same grid presented earlier. Since they have no conservative form, the numerical solution is obtained using the finite difference scheme proposed in Ref. 12.

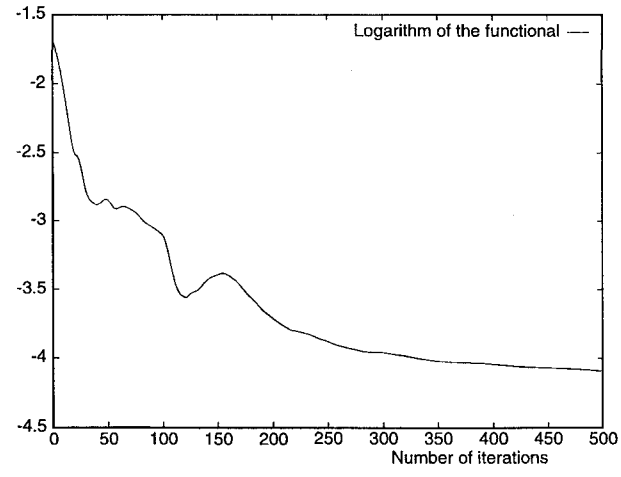


**Fig. 3 First optimization experiments. Duct with a bump. Logarithm of the functional $\mathcal{E}$ vs the number of iterations on the entire field.**

The computations are performed on a $40 \times 20$ grid. Total pressure and total temperature at the inlet are taken to be unity and $\sigma(0, y) = 0$. At the outlet the static pressure depends on the test case considered. For the lower wall ordinate $y(\Theta)$ we have

$$y(\Theta) = \begin{cases} 0 & \text{if} \quad -0.5 \le x < 0 \\ \displaystyle\sum_{i=1}^{4} \alpha_i\, x^{i+1}\,(x-1)^2 & \text{if} \quad 0 \le x < 1 \\ 0 & \text{if} \quad 1 \le x < 1.5 \end{cases}$$

This computational setup is identical to that in Ref. 12, and it represents a duct with a bump. We try to recover the pressure distribution obtained with the Euler solver, corresponding to the set of shape coefficients $\alpha = (0, 2, 2, 0)$. This means that the functional is 0 at the minimum. The outlet pressure is such that the flow presents a relevant shock at recompression.

Figure 3 shows the functional values at the end of the iterations on $\Omega$ of state equations (step 2 of the pseudotime algorithm). Figure 4 shows the convergence history of the state equation, computed to second-order accuracy, and the convergence history of the costate equation. In this figure and in all of the following, the residuals plotted are the rms of the first component of $\boldsymbol{U}_t$ for state and of $\boldsymbol{\Lambda}_t$ for costate. Finally, in Fig. 5, we present the starting pressure distribution and the one obtained at the end of the optimization procedure.

The practicability of this approach depends on the rate of convergence to the minimum. In fact, the state and costate equations converge to the steady solution with a less favorable rate compared with that of a simple analysis. It is easily seen that, because the shape is changing, the flowfield must change accordingly and so must the residuals. Figure 6 shows a comparison of the residuals for the state equations in the case of a simple analysis to the residuals in the optimization case. The convergence rate is three to four times slower in the optimization case. Considering the costate equations, the cost of the optimization procedure turns out to be on the order of 10 analysis; using the first of the two algorithms presented in Sec. IV, the factor of proportion is 100–200 depending on the updating strategy used. The CPU time needed on a DEC 3000/500 is 18 min with the algorithm presented. For the first algorithm of Sec. IV, 6 h of CPU time were needed.

The present rate of convergence could be improved by changing the way of updating the grid. In fact, close to the minimum, the entire grid is perturbed to update only the boundary. We believe that, close to the minimum, the rate of convergence can be improved by updating only the boundary points of the grid. In fact, the small difference between the desired pressure and the one obtained exists because close to the minimum the convergence rate of the equations is reduced. Therefore the pressure $p^*$ is obtained asymptotically.

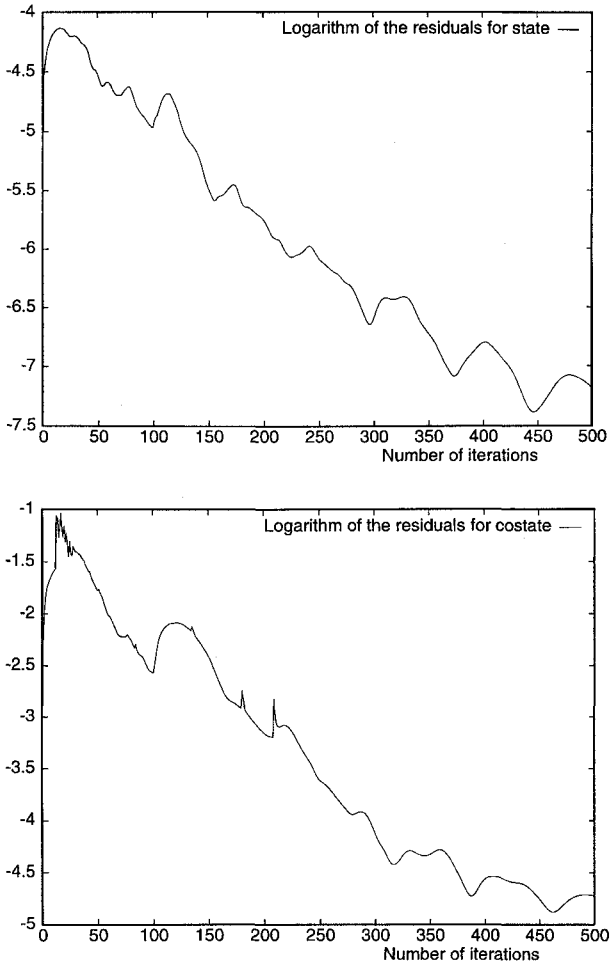


**Fig. 4 First optimization experiments. Duct with a bump. Convergence history: logarithm of the residuals of state and costate equations.**

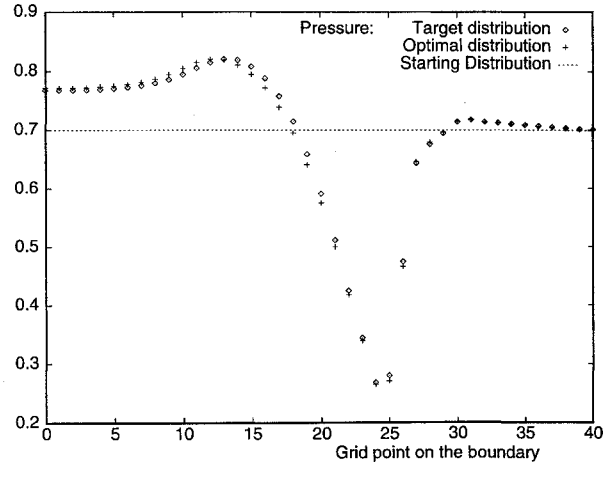


**Fig. 5 First optimization experiments. Duct with a bump. Target pressure distribution and optimal one. Starting pressure: constant distribution at 0.7 reference value.**

The computational cost breakdown for this application of the pseudotime method is found to be for the interior field (state and costate)

$$\frac{FLD\,m}{FLD\,m + BOU\,k}$$

and for the boundary (state and costate)

$$\frac{BOU\,k}{FLD\,m + BOU\,k}$$

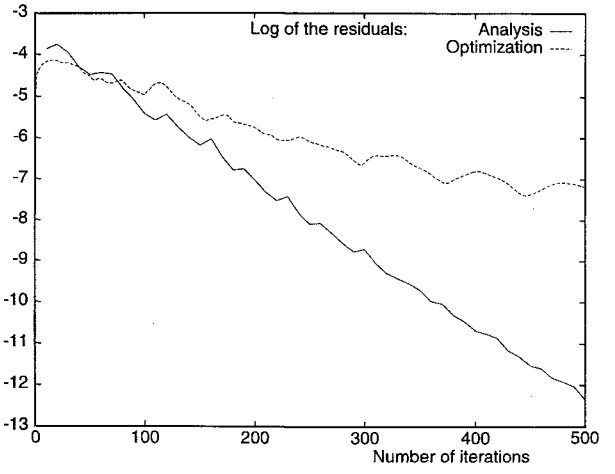


**Fig. 6 First optimization experiments. Duct with a bump. Convergence history of state equations in a simple analysis compared with the convergence of the state equations in an optimization procedure.**

where $m = 20$, $FLD$ is the number of iterations (10) for state and costate at steps 2 and 3 of the pseudotime algorithm, $k$ is the number of optimization steps on the boundary, typically 3, and $BOU$ is the number of iterations of state and costate on the boundary, typically 20. Therefore, for our cases, about 75% of the cost is due to the computations in the interior field and the rest to the optimization on the boundary.

## VI. Optimal Shape for Compressible Flows

The following examples represent situations for which the optimal solution is not generated with the same algorithm we used to study the optimization problem. In the first case, we recover a pressure distribution known theoretically and compare the shape obtained with the theoretical one.

The Ringleb flow[22] is a two-dimensional steady compressible isentropic flow, where subsonic, transonic, and supersonic regions are represented. It describes a 180-deg turn of a compressible flow; all of the exact values of the flow properties are given by simple formulas dependent on the stream function and on the Mach number. We consider the portion of the flow confined between two streamlines, which may be regarded as solid walls. The maximum Mach number on the bottom streamline is 1.6 and the minimum 0.8. On the top streamline the maximum Mach number is 0.8. These values determine uniquely the upper and the lower wall and the flow therein. The theoretical Mach number isocontours for such a flow are shown in Fig. 7.

The Ringleb pressure distribution on the bottom wall is taken as the desired distribution $p^* =$ that we want to achieve. The objective function is Eq. (4) with the preceding position.

The lower wall is described by the following parameterization:

$$y(\Theta) = r_0 + (r_1 - r_0) \sum_{i=1}^{4} \alpha_i \sin\left( i\,\pi\,\frac{\theta - \theta_0}{\theta_1 - \theta_0} \right)$$

where $r_0$ is the distance, measured from the point of intersection of the lines from the inlet and the outlet, to the first point on the lower wall, and $r_1$ is the distance to the last point. The angles $\theta_0$ and $\theta_1$ are relative to the first and last point, respectively, and are measured from the line from the inlet.

In principle, if we try to recover the pressure distribution on the lower wall, the solution is out of the design space in the sense that the functional will not be null at the minimum. In fact, the parameterization that we have chosen is not general but constrains the solution in a given subspace. We do not have any a priori knowledge of the values that the $\alpha_i$ will assume and how close to the desired pressure we can get.

In Fig. 8 it is seen that no visible difference can be appreciated between the theoretical wall shape and the optimal shape found. The points representing the two solutions do not overlap because they are computed on two different grids. In this case the functional (4)

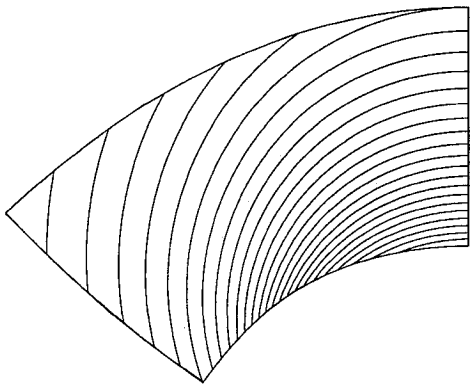

**Fig. 7 Ringleb flow. Mach number isocontours. The upper and the lower walls represent two streamlines of a compressible flow making a 180-deg turn.**

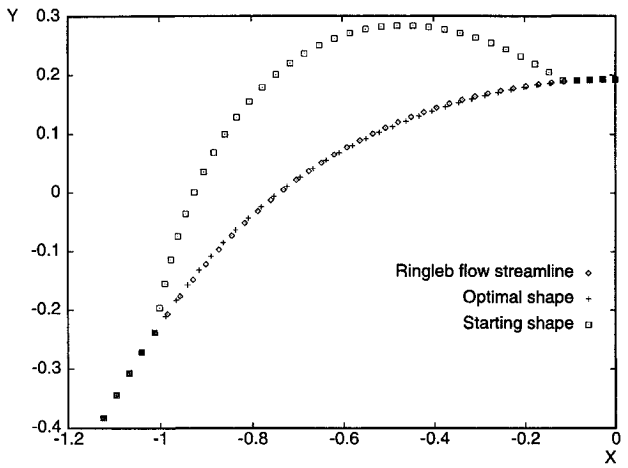


**Fig. 8 Ringleb flow: starting configuration, theoretical solution and optimal shape.**

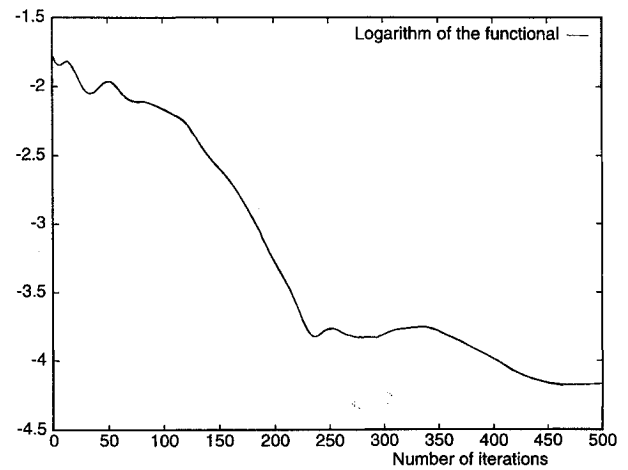


**Fig. 9 Ringleb flow: logarithm of the functional $\mathcal{E}$ vs the number of iterations on the entire field.**

is $6.70 \times 10^{-5}$ after 500 iterations of steps 2 and 3 of the second algorithm proposed. See Fig. 9. The CPU time required for this case is about 20 min.

In the second case considered, we are concerned with a convergent nozzle. The lower wall is represented by a parameterization similar to that given earlier, in which the parameters have been adapted to the present configuration. The inlet Mach number is 2.2, and the grid is $80 \times 40$. The optimization problem that we want to solve is to find a shape for the lower wall of the nozzle that allows a smooth compression. To this end we took as the desired pressure

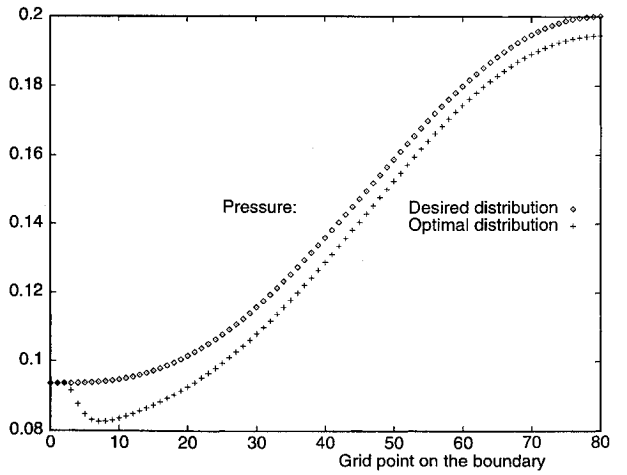


**Fig. 10 Convergent nozzle. Desired pressure distribution and optimal one. The optimal distribution and the desired one differ because of the parameterization of the lower wall chosen.**

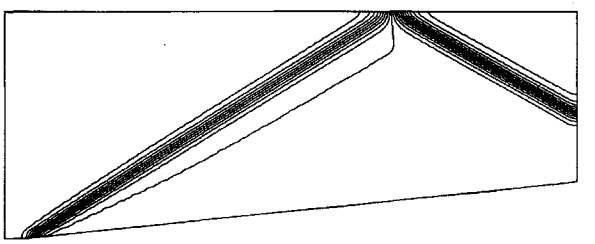

**Fig. 11 Convergent nozzle starting configuration. Pressure isocontours. Because of a supersonic inlet a shock is generated at the ramp. Subsequently the shock is reflected by the upper wall.**

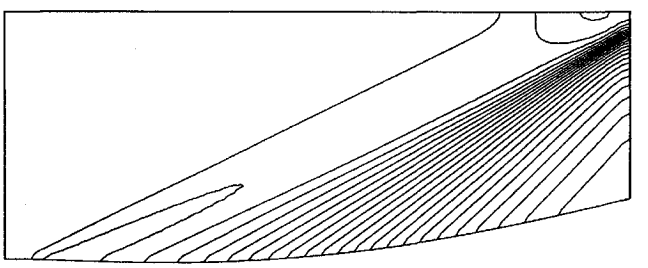

**Fig. 12 Convergent nozzle optimal solution. Pressure isocontours. A shockless compression is obtained.**

distribution on the lower wall $p^*$ that shown in Fig. 10. The starting configuration with pressure contourlines is shown in Fig. 11.

A relevant shock is present in the flowfield, and our objective is to eliminate it by requiring a smooth compression at the lower wall. The smooth pressure distribution is not perfectly attained, as is seen in Fig. 10. Nevertheless, the compression appears to be smooth, and the shock is eliminated from the flowfield (Fig. 12). These results are obtained after 200 iterations in about 35 min of CPU time. The functional is decreased from $3.75 \times 10^{-3}$ to $1.35 \times 10^{-4}$. The computation has been pursued for 2000 iterations and the functional value remained unchanged. The fact that the optimal pressure distribution is not attained perfectly is a result of the kind of parameterization that is used and the number of design variables used. Moreover, it is not assured that the pressure distribution chosen is compatible with a shockless compression; therefore we do not know whether the target pressure distribution can be attained exactly even using an infinite dimensional design space.

Finally, an experiment using eight shape coefficients is performed. In Fig. 13 it is seen that the convergence rates of the state and costate equations are affected only in the first iterations. The functional minimum is therefore attained with the same number of iterations as in the case of four shape coefficients.

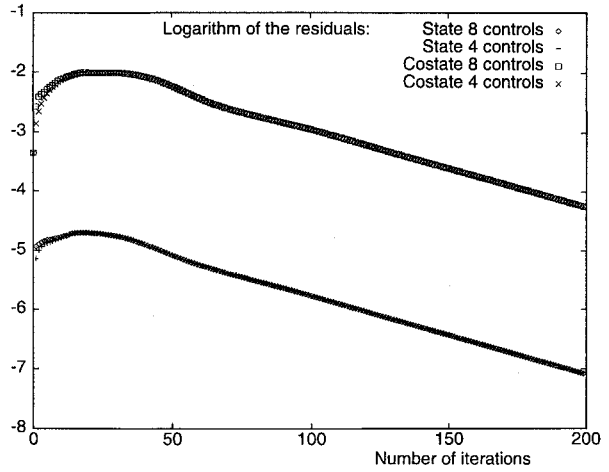


**Fig. 13 Convergent nozzle. State and costate convergence history: four shape coefficients vs eight shape coefficients. The convergence differs only in the beginning steps.**

## VII. Conclusions

The pseudotime method was applied to optimization problems governed by the Euler equations in two dimensions. The problem of matching the pressure distribution to a desired one was considered, both for subsonic and supersonic flows. The rate of convergence to the minimum for the cases considered is three to four times slower compared with that of the analysis problem. Results were obtained for Ringleb flow and a shockless compression case. The algorithm could be implemented with no substantial changes to existing adjoint-based optimization codes. Numerical results indicate that the method converges at a rate that is independent of the number of design variables. The method offers a powerful and inexpensive tool for the study of nonintuitive configurations for aerodynamic design.

## Acknowledgments

This research was supported in part under NASA Contract NAS1-19480. The first author, who started this research at the Institute for Computer Applications in Science and Engineering, would like to thank Manuel D. Salas of NASA Langley and Maurizio Pandolfi of the Politecnico di Torino for stimulating discussions that contributed to the formulation of the algorithm proposed.

## References

[1]Ta'asan, S., "Pseudo-Time Methods for Constrained Optimization Problems Governed by PDE," NASA CR-195081, May 1995.

[2]Lighthill, M. J., "A New Method of Two Dimensional Aerodynamic Design," ARC, Rand M 2112, 1945.

[3]Bauer, F., Garabedian, P., and Korn, D., *Supercritical Wing Sections*, Springer–Verlag, Berlin, 1972.

[4]Zannetti, L., "A Natural Formulation for the Solution of Two-Dimensional or Axisymmetric Inverse Problems," *International Journal of Numerical Methods in Engineering*, Vol. 22, 1986, pp. 451–463.

[5]Chaviaropoulos, P., Dedoussis, V., and Papailiou, K. D., "Single-Pass Method for the Solution of the Inverse Potential and Rotational Problems, Part ii: Fully 3-D Potential Theory and Applications," AGARD Rept. 803, Nov. 1994, pp. 1-1–1-19.

[6]Lions, J. L., *Optimal Control of Systems Governed by Partial Differential Equations*, Springer–Verlag, Berlin, 1971.

[7]Pironneau, O., "On Optimum Profiles in Stokes Flow," *Journal of Fluid Mechanics*, Vol. 59, Pt. 1, 1973, pp. 117–128.

[8]Pironneau, O., "On Optimum Design in Fluid Mechanics," *Journal of Fluid Mechanics*, Vol. 64, Pt. 1, 1974, pp. 97–110.

[9]Glowinski, R., and Pironneau, O., "On the Numerical Computation of the Minimum-Drag Profile in Laminar Flow," *Journal of Fluid Mechanics*, Vol. 72, Pt. 2, 1975, pp. 385–389.

[10]Jameson, A., "Aerodynamic Design Via Control Theory," NASA CR-181749, Nov. 1988.

[11]Iollo, A., Salas, M. D., and Ta'asan, S., "Shape Optimization Governed by the Euler Equations Using an Adjoint Method," NASA CR-191555, Nov. 1993; see also *14th International Conference on Numerical Methods in Fluid Dynamics*, Lecture Notes in Physics 453, Springer–Verlag, Berlin, 1995, pp. 274–279.

[12]Iollo, A., and Salas, M. D., "Contribution to the Optimal Shape Design of Two-Dimensional Internal Flows with Embedded Shocks," NASA CR-195062, March 1995; see also *Journal of Computational Physics*, Vol. 125, June 1996, pp. 124–134.

[13]Beux, F., and Dervieux, A., "Exact-Gradient Shape Optimization of a 2D Euler Flow," *Finite Elements in Analysis and Design*, Vol. 12, 1992, pp. 281–302.

[14]Rizk, M. H., "Optimizing Advanced Propeller Designs by Simultaneously Updating Flow Variables and Design Parameters," *AIAA Journal*, Vol. 26, No. 6, 1989, pp. 515–522.

[15]Ta'asan, S., "One Shot Methods for Optimal Control of Distributed Parameters Systems," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 91-2, Hampton, VA, Jan. 1991.

[16]Ta'asan, S., Kuruvila, G., and Salas, M. D., "Aerodynamic Design and Optimization in One Shot," AIAA Paper 92-005 Jan. 1992.

[17]Arian, E., and Ta'asan, S., "Multigrid One Shot Methods for Optimal Control Problems: Infinite Dimensional Control," NASA CR-194939, July 1994.

[18]Beux, F., and Dervieux, A., "A Hierarchical Approach for Shape Optimization," Institut National de Recherche en Informatique et en Automatique, INRIA Rept. 1868, Sophia Antipolis, France, Feb. 1993.

[19]Jameson, A., "Optimum Aerodynamic Design Via Boundary Control," AGARD Rept. 803, Nov. 1994, pp. 3-1–3-33.

[20]Pandolfi, M., "A Contribution to the Numerical Prediction of Unsteady Flows," *AIAA Journal*, Vol. 22, No. 5, 1984, pp. 602–610.

[21]Harten, A., Engquist, B., and Chakravarthy, S. R., "Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, iii," *Journal of Computational Physics*, Vol. 71, Aug. 1987, pp. 231–303.

[22]Ringleb, F., "Exakte Loesungen der Differentialgleichungen einer adiabatischen Gasstroemung," *ZAMM*, Vol. 20, 1940.