JOURNAL OF COMPUTATIONAL PHYSICS 117, 67–78 (1995)

High Order Centered Difference Methods for the Compressible Navier–Stokes Equations

BJÖRN SJÖGREEN

Department of Scientific Computing, Uppsala University, S-751 04 Uppsala, Sweden

Received February 22, 1994

We study centered finite difference methods of general order of accuracy $2p$. Boundary points are approximated by one sided operators. We give boundary operators which are stable for the linear advection equation. In cases where the approximation is unstable, we show how stability can be recovered by use of high order artificial dissipation operators. The methods are generalized to the compressible Navier–Stokes equations. We obtain a highly accurate grid converged finite difference solution of the Navier–Stokes equations, which we use to evaluate the accuracy of a finite volume TVD shock capturing method.

1. INTRODUCTION

In this report we develop finite difference methods of high order and use them in practical computations of compressible Navier–Stokes flows. The motivation for this work is twofold. First, we want to investigate the gain in efficiency from using high order methods. On simple model problems, this gain is substantial [5]. The question is whether this is true also for more complicated problems. Second, a good centered difference high order method can be used to produce a reference solution for convergence studies. In Section 6 we will show a comparison of results from a TVD shock-capturing method and see how that solution converges to the "exact" high order finite difference solution. Finally, a good high order difference method can be used for direct simulation of turbulence.

The philosophy of this work is to cover all the steps from the stability analysis on a linear scalar problem to the implementation for the 2D Navier–Stokes system. In this way we will understand the reliability of the linear scalar stability analysis and what additional difficulties are introduced by going to more space dimensions and systems of equations.

It is well known [5] that operators of higher orders of accuracy are more efficient. The drawbacks of high orders are

(a) strong assumptions about the regularity of the solution;

(b) wide stencils call for many numerical boundary conditions.

The problem with regularity (a) can be resolved using modern ENO methods. In this report we shall instead address point (b).

2. STABLE HIGH ORDER METHODS

The goal is to develop high order methods which are stable and easy to implement. We want to solve the Navier–Stokes equations of compressible fluid flow, which are a system of PDE on the form

$$\mathbf{w}_t + \mathbf{f}(\mathbf{w})_x + \mathbf{g}(\mathbf{w})_y = \mathbf{f}_v(\mathbf{w},\mathbf{w}_x, \mathbf{w}_y)_x + \mathbf{g}_v(\mathbf{w},\mathbf{w}_x, \mathbf{w}_y)_y.$$

As a first step we define and analyze numerical methods for the scalar model problem,

$$u_t + au_x = 0, \quad 0 \le x < \infty\ t > 0$$
$$u(0, x) = u_0(x)$$
$$u(t, 0) = u_b(t) \quad \text{if } a > 0.$$

We discretize the computational domain, $0 \le x < \infty$, by the uniform grid $x_j = (j - 1)\,\Delta x$, $j = 1, 2, \ldots$, where Δx is the distance between the grid points.

The x-derivative will be discretized by a pth-order accurate centered difference operator

$$D_p v_j = \frac{1}{\Delta x} \sum_{k=-s}^{s} \alpha_k^p v_{j+k} = u_x(x_j) + O(\Delta x^p).$$

It is assumed that $p = 2q$, q integer. The coefficients for the second- and fourth-order methods are

$$\alpha_{-1}^2 = -\tfrac{1}{2}, \quad \alpha_0^2 = 0, \quad \alpha_1^2 = \tfrac{1}{2},$$
$$\alpha_{-2}^4 = \tfrac{1}{12}, \quad \alpha_{-1}^4 = -\tfrac{2}{3}, \quad \alpha_0^4 = 0, \quad \alpha_1^4 = \tfrac{2}{3}, \alpha_2^4 = -\tfrac{1}{12}.$$

Coefficients for a general q can be found in [1].

We settle for using a Runge–Kutta method in time, and thus we concentrate on the semi-discrete problem,

$$\frac{dv_j(t)}{dt} = -aD_p v_j(t), \quad j = q + 1, q + 2, q + 3, \ldots.$$

0021-9991/95 $6.00

The centered difference operator is $p + 1$ points wide, and thus requires $q = p/2$ extra boundary conditions. At the boundary points we approximate $u_x(x_j)$ by pth-order accurate one-sided difference operators, with the stencil extending over the grid points 1, 2, ..., m,

$$D_{pb}^j v_j = \frac{1}{\Delta x} \sum_{r=1}^{m} \beta_{j,r}^p v_r.$$

The total approximation is thus

$$\begin{aligned} \frac{dv_j(t)}{dt} &= -aD_{pb}^j v_j(t), \quad j = 1, ..., q, \\ \frac{dv_j(t)}{dt} &= -aD_p v_j(t), \quad j = q+1, q+2, q+3, ..., \end{aligned} \tag{2.1a}$$

if $x = 0$ is an outflow boundary and

$$\begin{aligned} v_1(t) &= u_b(t), \\ \frac{dv_j(t)}{dt} &= -aD_{pb}^j v_j(t), \quad j = 2, ..., q, \\ \frac{dv_j(t)}{dt} &= -aD_p v_j(t), \quad j = q+1, q+2, q+3, ..., \end{aligned} \tag{2.1b}$$

if $x = 0$ is an inflow boundary.

The most straightforward boundary operator to use is the smallest possible stencil which maintains the order of accuracy. For pth-order accuracy, we approximate the derivatives at the points $x_1, ..., x_q$, by using $x_1, ..., x_{p+1}$ for the stencil of the operator.

As an example, consider fourth-order accuracy in the interior. We need two boundary operators to approximate $u_x(x_1)$ and $u_x(x_2)$. We use the grid points $(x_1, ..., x_5)$ as stencil for the fourth-order boundary approximations. The boundary operators are

$$\begin{aligned} D_{4b}^1 v_1 &= (-3v_5 + 16v_4 - 36v_3 + 48v_2 - 25v_1)/(12\,\Delta x) \\ D_{4b}^2 v_2 &= (v_5 - 6v_4 + 18v_3 - 10v_2 - 3v_1)/(12\,\Delta x). \end{aligned}$$

We will also investigate another set of boundary operators which are obtained as above, but with a stencil which is one point wider. This extra degree of freedom is used to match the leading error term in the Taylor expansion of the approximation error of the boundary operator with the leading term in the error of the interior operator. If we have for the interior approximation

$$D_p v_j = u_x(x_j) + c_p u^{(p+1)} \Delta x^p + O(\Delta x^{p+1}),$$

we determine the boundary operators such that

$$D_{pe}^j v_j = u_x(x_j) + c_p^j u^{(p+1)} \Delta x^p + O(\Delta x^{p+1})$$

with $c_p^j = c_p$. For the case of fourth-order interior approximation, this leads to

$$\begin{aligned} D_{4e}^1 v_1 &= (2v_6 - 13v_5 + 36v_4 - 56v_3 + 58v_2 - 27v_1)/(12\,\Delta x) \\ D_{4e}^2 v_2 &= (-v_6 + 6v_5 - 16v_4 + 28v_3 - 15v_2 - 2v_1)/(12\,\Delta x). \end{aligned}$$

We denote these error matched boundary conditions by D_{pe}^j. The reason for this type of boundary operator is that for the Navier–Stokes equations, the operators will be applied twice to approximate second derivatives. Thus the leading error term will be differentiated by the second application of the approximation. Unless the leading error term is continuous, loss of at least one order of accuracy will occur.

3. STABILITY ANALYSIS

In this section we analyze the stability of the test problems (2.1a) and (2.1b). We analyze the inflow ($a > 0$) case and the outflow ($a < 0$) case separately, but we do not here consider hyperbolic *systems* of equations, although a system could be analyzed similarly (for one space dimension), by decoupling it into characteristic variables. The analysis is done by numerically evaluating the eigenvalues and generalized eigenvalues of the difference approximation. We describe below some main features of this method of stability analysis. Similar automatic stability analysis has been done (with more generality than here) in [10]. However, [10] treats the fully discretized problem, which becomes very complicated when multi-stage Runge–Kutta methods are employed.

We recall the stability theory for initial-boundary value problems (IBVP) as described in [3]. The analysis used here is the semi-discrete version of [3]. Although we have followed the description given in a forthcoming book [4], the ideas presented here are very similar to [3] and can easily be understood from [3].

Consider the general IBVP,

$$\begin{aligned} \frac{dv_j(t)}{dt} &= Qv_j(t) + F_j, \quad j = 1, 2, ..., \\ L_0 \begin{pmatrix} v_{-q+1}(t) \\ ... \\ v_m(t) \end{pmatrix} &= g(t), \end{aligned} \tag{3.1}$$

where Q is the difference operator in the interior and the boundary conditions are represented by the operator L_0, the grid points up to x_m are involved in the boundary procedure. The problem is defined on $j = 1, 2, ...$ The approximation with boundary operators,

$$\frac{dv_j(t)}{dt} = Qv_j(t) + F_j, \quad j = q+1, q+2, ...,$$

$$\frac{dv_j(t)}{dt} = Q_b^j v_j(t) + F_j, \quad j = 1, 2, ..., q,$$

can be put in the form (3.1), with $g(t) = 0$, by writing it as

$$\frac{dv_j(t)}{dt} = Qv_j(t) + F_j, \quad j = 1, 2, ...,$$

$$(Q_b^j - Q)v_j(t) = 0, \quad j = 1, 2, ..., q,$$

where we now have introduced the extra variables $v_{-q+1}(t)$, ..., $v_0(t)$. Thus the operator L_0 is in this case constructed from $Q_b^j - Q$.

In [4], the problem (3.1) is defined as stable if an estimate

$$\int_0^\infty e^{-2\eta t}\|v(t)\|_h^2\, dt \leq K(\eta) \int_0^\infty e^{-2\eta t}\|F(t)\|_h^2\, dt$$

can be found for Δx sufficiently small, and for all $\eta > \eta_0$, and with $K(\eta) \to 0$ when $\eta \to \infty$.

We next show how to investigate whether a method is stable or not according to this definition. We only give the necessary steps, without explaining why they are done. We refer to [3, 4], for motivations and proofs of the statements below.

In order to investigate the stability, we Laplace transform the problem (3.1) in time by introducing $v_j(t) = e^{st}\hat{v}_j$. (3.1) becomes

$$\tilde{s}\hat{v}_j = \tilde{Q}\hat{v}_j + \tilde{F}_j$$

$$L_0\hat{v} = \hat{g}$$

$$\|\hat{v}\| < \infty,$$

where $\tilde{s} = s\,\Delta x$, and $\tilde{Q} = \Delta x Q$. This is an eigenvalue problem. If it has a solution with $\mathrm{Re}\,(s) > 0$, then the difference method is not stable, since the factor e^{st} then permits unlimited growth of the solution. It is possible to derive the properties of this problem from the homogeneous problem

$$\tilde{s}\hat{v}_j = \tilde{Q}\hat{v}_j \tag{3.2a}$$

$$L_0\hat{v} = 0 \tag{3.2b}$$

$$\|\hat{v}\| < \infty. \tag{3.2c}$$

Equation (3.2a) is a linear difference equation, which we solve by standard methods. Since we use centered differences in the interior, $Qv_j = \sum_{k=-q}^{q} \alpha_k v_{j+k}$, the characteristic polynomial is given by

$$\sum_{k=0}^{2q} \alpha_{k-q} r^k - sr^q = 0.$$

First we compute the roots of this equation, r_1, r_2, ..., r_{2q}. Because of condition (3.2c), we can disregard the roots with modulus greater than one. From the stability of the periodic problem, it follows that there are exactly q roots for $\mathrm{Re}\,(s) > 0$; see [3]. If there are no multiple roots we have the solution

$$\hat{v}_j = c_1 r_1^j + c_2 r_2^j + \cdots + c_q r_q^j, \tag{3.3}$$

where the remaining constants c_1, ..., c_q are determined from the boundary conditions.

Remark. If $r_1 = r_2$ is a double root, we get terms of the form $(c_1 + jc_2)r_1^j$ in (3.3). In a computer program this is handled by substituting $c_1 r_1^j + c_2 r_2^j$ by

$$c_1 r_1^j + c_2 \frac{r_1^{j+1} - r_2^{j+1}}{r_1 - r_2},$$

if r_1 and r_2 are closer than a given tolerance. The quotient above converges to jr_1^j when $r_1 \to r_2$. Similar expressions can be found for roots of higher multiplicity.

Inserting (3.3) into the boundary condition $L_0\hat{v} = 0$ yields a linear system of equations

$$C(\tilde{s}) \begin{pmatrix} c_1 \\ c_2 \\ \cdots \\ c_q \end{pmatrix} = 0,$$

where the matrix C is a function of $\tilde{s}$, and from which the stability can be practically investigated according to the following theorem.

THEOREM 3.1. *If there are no roots of the equation*

$$\mathrm{Det}(C(\tilde{s})) = 0$$

with $\mathrm{Re}(\tilde{s}) \geq 0$ *then the approximation* (3.1) *is stable.*

For a proof, see [3, 4].

It is now possible to perform this stability investigation numerically. Given $\tilde{s}$, go through the following steps to find $\mathrm{Det}(C)$.

1. Form the characteristic polynomial and find its roots.
2. Remove the roots which are greater than one.
3. Form the matrix C from the boundary conditions.
4. Form the determinant.

It is simple to implement this, since standard library routines can be used to calculate the roots of the polynomial and to find the determinant.

Once the function $\mathrm{Det}(C(\tilde{s}))$ can be evaluated, it is straightforward to solve the 2×2 system

TABLE I

Roots of $\mathrm{Det}(C(\tilde{s}))$ in the Right-Hand Plane

Interior	Boundary	Inflow	Outflow	Stable
D_4	D_{4b}	No roots	No roots	Yes
D_4	D_{4e}	$\tilde{s} = 0.093 + 1.308i$	No roots	No
D_6	D_{5b}	$\tilde{s} = 0.058 + 1.337i$	No roots	No
D_8	D_{8b}	$\tilde{s} = 0.417 + 1.443i$	No roots	No
D_8	D_{8e}	$\tilde{s} = 0.496 + 1.478i$	No roots	No

$$\mathrm{Det}(C(\tilde{s})) = 0, \tag{3.4}$$

using Newton's method with numerical evaluation of the Jacobian matrix.

Remark. It is possible to substitute the determinant above with some other more numerically well-conditioned quantity which measures the singularity of the system, e.g., the smallest singular value.

Remark. When $\mathrm{Re}(s) \approx 0$, there is a risk that we will get the wrong number of roots, since some roots will have absolute value $=1$ to machine precision. In such cases we evaluate the derivative dr/ds by implicit differentiation of the characteristic equation. From the direction of this derivative we easily check whether a root "belongs" to our solution in the right-hand plane or not.

The computer program can plot the surface $\mathrm{Det}(C(\tilde{s}))$ in the plane $\mathrm{Re}(\tilde{s}) \geq 0$, and it can solve (3.4) using a Newton like method. To be sure to be within the domain of convergence for Newton's method, we discretize a sufficiently large part of $\mathrm{Re}(s) \geq 0$ and start a Newton iteration from each grid point. If the method converges to a root we stop the program and return the root.

The stability analysis was tried out on some cases, with result shown in Table I. The boundary operators are as described in Section 2. D_{pb} has the smallest stencil of order p, and D_{pe} is the error matched boundary operator of order p. In the inflow case a boundary value was given at x_1 and the boundary operators used at x_2, ..., x_q. For the outflow case the boundary operators were used also at x_1.

TABLE II

The Necessary Artificial Dissipation to Make the Method Stable

Interior	Boundary	Dissipation, $4q$	Stable for
D_4	D_{4e}	8	$\hat{d} > 0.0029$
D_6	D_{5b}	8	$\hat{d} > 0.0018$
D_6	D_{5b}	12	$\hat{d} > 0.0013$
D_8	D_{8b}	12	$\hat{d} > 0.00076$
D_8	D_{8e}	12	$\hat{d} > 0.00092$

We conclude that the method (D_4, D_{4b}) is stable.

A complete listing from the root search for approximations of type (D_p, D_{rb}), $r \leq p$, is given in Appendix A.

The result from stability analysis was confirmed by solving the test problem (2.1) on a bounded domain with inflow on one boundary and outflow on the other. We show in the next section how the unstable methods can be made stable by using artificial dissipation.

4. SEMI-DEFINITE ARTIFICIAL DISSIPATION OPERATORS

In this section we improve on the stability of the high-order centered-difference methods by adding artificial dissipation. Furthermore, artificial dissipation is necessary for solving nonlinear problems such as the compressible Navier–Stokes equations, due to nonlinear instabilities.

We will study approximations of the type

$$\frac{dv_j}{dt} = -aD_p v_j - \frac{d}{\Delta x}(\Delta_+ \Delta_-)^{2q} v_j, \tag{4.1}$$

where thus a $4q$th derivative is used as artificial dissipation. This approximation will be denoted (D_p, D_{pb}, V_{4q}), where D_{pb} is the boundary difference approximation operator. We have added the general dissipation operator

$$-\frac{d}{\Delta x}(\Delta_+ \Delta_-)^{2q} u_j$$

to the approximation in Section 3. Before performing a stability analysis on the problem (4.1), we study the dissipation operator separately. For $q = 1$ this is the well-known fourth-order dissipation operator, often used in CFD codes. The operator restricts the accuracy to at most $4q - 1$.

If the operator is used with periodic boundary conditions, it is semi-definite in the usual scalar product

$$(u, v) = \sum_{j=1}^{N} \bar{u}_j v_j \Delta x.$$

This follows from the rule for periodic grid functions, u,

$$(u, \Delta_+ u) = -(\Delta_- u, u)$$

TABLE III

L^∞ Norm of Truncation Errors

Grid	D_8/M_4	D_8/M_8	D_4/M_4	D_4/M_2
G_1	7.2×10^{-3}	2.5×10^{-3}	6.2×10^{-3}	0.97
G_2	9.74	4.2×10^{-2}	8.7×10^{-2}	30.4

from which follows

$$(u, (\Delta_+\Delta_-)^{2q}u) = ((\Delta_+\Delta_-)^q u, (\Delta_+\Delta_-)^q u) \geq 0.$$

This property is normally used to show that the forward in time problem

$$\frac{du_j(t)}{dt} = -(\Delta_+\Delta_-)^{2q}u_j(t)$$

is well-posed. Take the scalar product by u, and we obtain

$$\frac{d\|u\|^2}{dt} = -(u, (\Delta_+\Delta_-)^{2q}u) \leq 0.$$

Thus the norm of u decreases in time.

Next we modify the dissipation operator when the boundaries are not periodic, as described in [7]. It is possible to keep the semi-definiteness at the price of reducing the order of accuracy at the boundaries.

Consider the splitting of the periodic operator into an interior part and a boundary part,

$$(\Delta_+\Delta_-)^q = (\Delta_+\Delta_-)_I^q + (\Delta_+\Delta_-)_B^q,$$

where $((\Delta_+\Delta_-)_I^q v_j) \cdot ((\Delta_+\Delta_-)_B^q v_j) = 0$. For example, $q = 1$,

$$\begin{pmatrix} -2 & 1 & 0 & 0 & \dots & 1 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & \dots & & & \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 1 & \dots & 0 & 0 & 1 & -2 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ & & \dots & & & \\ 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & \dots & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$+ \begin{pmatrix} -2 & 1 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ & & \dots & & \\ 0 & \dots & 0 & 0 & 0 \\ 1 & \dots & 0 & 1 & -2 \end{pmatrix}.$$

Define the dissipation operator for a problem with boundaries as

$$(\Delta_+\Delta_-)_s^{2q} = (\Delta_+\Delta_-)^q(\Delta_+\Delta_-)_I^q.$$

The interior operator $(\Delta_+\Delta_-)_I^q$ has zeros on the boundary and will destroy all periodic terms [7]. The resulting operator is thus local also on the boundary, and it has the semi-definite property

$$\begin{aligned} -((\Delta_+\Delta_-)^q(\Delta_+\Delta_-)_I^q v_j, v_j) &= -((\Delta_+\Delta_-)_I^q v_j, (\Delta_+\Delta_-)^q v_j) \\ &= -((\Delta_+\Delta_-)_I^q v_j, ((\Delta_+\Delta_-)_I^q \\ &\quad + (\Delta_+\Delta_-)_B^q)v_j) \\ &= -\|(\Delta_+\Delta_-)_I^q\|^2 \leq 0. \end{aligned}$$

The operators are stable, but the accuracy on the boundary is reduced, according to

$q = 1 \Rightarrow$ accuracy interior = 3, on boundary = 1

$q = 2 \Rightarrow$ accuracy interior = 7, on boundary = 3

$q = 3 \Rightarrow$ accuracy interior = 11, on boundary = 5.

We next perform the normal mode stability analysis from Section 3, with these artificial dissipation operators included; i.e., we consider the approximation (4.1). The dissipation operators are modified at the boundary as described above. We study stability as a function of the normalized viscosity parameter $\hat{d} = d/a$. For the unstable methods, it turns out that the root in the right-hand plane $\tilde{s}^*$ moves to the left as $\hat{d}$ increases, and at a certain point it leaves the right-hand plane, and thus the method becomes stable. Figure 4.1 shows the trajectory of the root in the $\tilde{s}$ plane as $\hat{d}$ increases from zero for the case (D_8, D_{8e}, V_{12}). Table II shows the necessary amount of dissipation to make the methods in Table I stable.

5. IMPLEMENTATION FOR THE 2D COMPRESSIBLE NAVIER–STOKES EQUATIONS

We now use the methods analysed in Sections 3 and 4 to solve the Navier–Stokes equations for compressible fluid flow on a curvilinear grid. The main difference compared with the model problem are the inclusion of metric derivatives and the use of second derivatives in the physical viscosity operator.

Assume that we are given the grid

TABLE IV

L^∞ Errors in Entropy on the Body for Grids of Various Refinements

Method	17 × 9	33 × 17	65 × 33	129 × 65
(D_2, D_{b2}, V_4)	5.2×10^{-2}	1.8×10^{-2}	6.7×10^{-3}	2.2×10^{-3}
(D_4, D_{b4}, V_8)	2.4×10^{-2}	6.5×10^{-3}	1.4×10^{-3}	3.8×10^{-4}

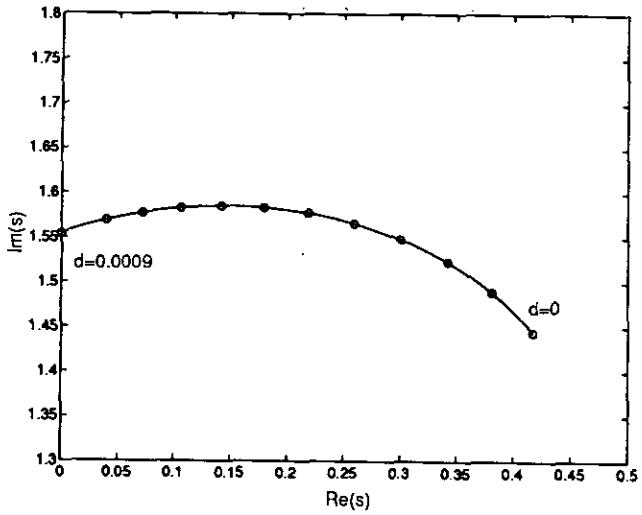

FIG. 4.1. Root of Det($C(\tilde{s})$) as function of $\hat{d}$.

$$\{(x(\xi, \eta), y(\xi, \eta) : (\xi, \eta) \in [1 \ldots N_x, 1 \ldots N_y]\}.$$

A system of conservation laws transforms into

$$\mathbf{w}_t + \frac{1}{\sqrt{g}}((y_\eta \mathbf{f}(\mathbf{w}) - x_\eta \mathbf{g}(\mathbf{w}))_\xi + (-y_\xi \mathbf{f}(\mathbf{w}) + x_\xi \mathbf{g}(\mathbf{w}))_\eta) = 0$$

in the uniform (ξ, η) space. Here the determinant of the Jacobian of the metric mapping is $\sqrt{g} = x_\xi y_\eta - x_\eta y_\xi$. The problem is now given on a uniform grid, so that the operators discussed in the previous sections can be applied directly in each coordinate direction.

The Navier–Stokes operator contains terms of the type

$$(a(u)u_x + b(u)u_y)_x. \tag{5.1}$$

These are approximated by using the high order formulas to find

$$u_x = \frac{1}{\sqrt{g}}(y_\eta u_\xi - y_\xi u_\eta)$$

$$u_y = \frac{1}{\sqrt{g}}(-x_\eta u_\xi - x_\xi u_\eta)$$

at each grid point (including boundary points) and then by inserting these into (5.1) and applying the same formulas again to approximate the outermost x-derivative. This means that; e.g., u_{xx} is approximated by $(D_p)^2 u$. In this way stencils become wide, but the method becomes very simple to implement.

To maintain the order of accuracy, the metric derivatives x_ξ, x_η, ... are evaluated using the same order of accuracy as for the flux derivatives. In all computations, exactly the same formulas are used for the metric derivatives as for the flux derivatives. Using the same difference operator leads to a small truncation error [9]. We have made some numerical experiments to verify that the truncation error has the right order. In this verification, the effect from the grid derivatives can be clearly seen. More specifically, we divide the equations into time derivative plus inviscid Euler terms and Navier–Stokes viscosity terms, according to

$$u_t + E(u) = NS(u).$$

We write formally the semi-discrete approximation as

$$\frac{dv_j}{dt} = -E_N(u) + NS_N(u),$$

where E_N and NS_N are the numerical approximations of the Euler and Navier–Stokes terms, respectively. Take now an arbitrary function $u(x, y)$ and compute

$$E(u) - E_N(u) \quad \text{or} \quad NS(u) - NS_N(u)$$

on grids of various refinement. This gives the truncation error for each of the terms. In Table III we show the L^∞ norm of the error in the Navier–Stokes terms on two different grids. G_1 is a very smooth grid with small variation in cell size. G_2 is an extremely stretched grid for boundary layer computations. D_p/M_q means that the order of accuracy in the difference approximation is p and the order of approximation of the metric derivatives is q.

If the grid is not regular enough, the way we treat the metric is very critical. On the other hand, if the grid transformation is smooth the effect of grid derivatives is only related to the order of accuracy in the approximation of the derivatives x_ξ, x_η, y_ξ, y_η.

6. NUMERICAL RESULTS

One of the objectives of this work was to evaluate the performance of TVD and ENO schemes for supersonic external flows. By using a very accurate centered-difference method, one can compute a very accurate solution, which can be used as reference solution for a shock capturing scheme.

First some comments about the validity of the linear stability analysis. The methods of order eight did not perform well on the two-dimensional Euler/Navier–Stokes equations. There is an additional instability emanating from the corners of the computational domain for the eighth-order method. Thus the results in this section are obtained only by second- and fourth-order methods. In all cases the artificial dissipation operators from Section 4 were necessary, even in the case (D_4, D_{4b}) which is linearly stable without artificial dissipation. The dissipation coefficient was scaled with the maximum spectral radius of the Jacobian of the Euler system, corresponding to the scaling $\hat{d} =$

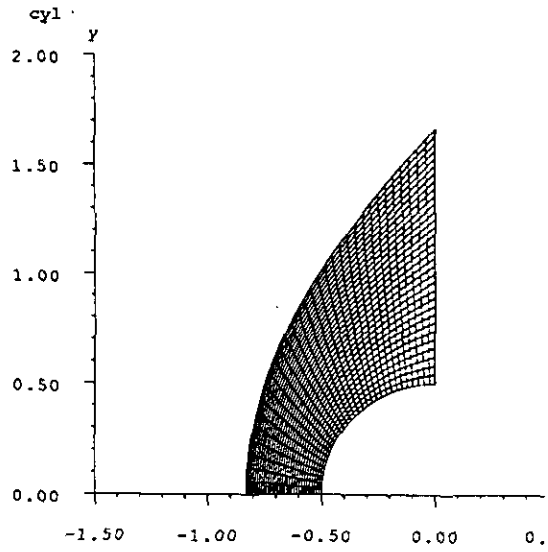

FIG. 6.1. Computational domain.

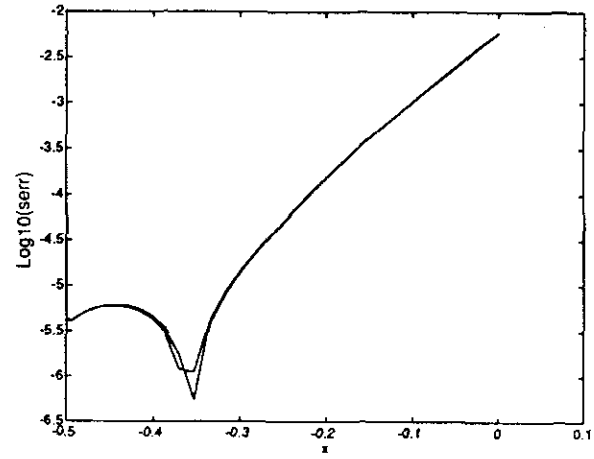

FIG. 6.3. Error in entropy along the body.

d/a in the scalar linear case (see Section 3). Even so, the stability limits derived in Section 4 for the coefficient $\hat{d}$, could not be used. In all cases the coefficients $\hat{d}$ had to be chosen larger than the lower stability limit derived in Section 4.

The test case we use is the supersonic flow past a disk. The grid has $n_i \times n_j$ points and is arranged such that the wall boundary is given by $(i, 1)$ and the outer boundary by (i, n_j). Figure 6.1 shows the computational domain.

The free stream Mach number is 3.15 and the Reynolds number is 1000. We give the temperature on the wall as 288K and use Sutherland's law to describe the temperature dependence of the viscosity. The centered difference methods cannot be used for discontinuous solutions. Instead we fit the outer boundary of the grid to the bow shock by using the Rankine–Hugoniot condition. Thus the influence of the Navier–Stokes viscosity is neglected at the outer boundary. The R-H condition normal to the boundary,

$$s(\mathbf{w}_\infty - \mathbf{w}_{i,n_j}) = \mathbf{f}(\mathbf{w}_\infty) - \mathbf{f}(\mathbf{w}_{i,n_j}),$$

gives four equations for the unknown shock speed s and the unknown solution vector in the outermost point $\mathbf{w}_{i,n_j}$. The free stream state $\mathbf{w}_\infty$ is given; hence, we have four equations for five unknowns. However, one characteristic is propagating from the interior of the domain into the bow shock, and by keeping this outflow characteristic unchanged we obtain a fifth equation which closes the system. Thus the grid velocity, s, is computed at $j = n_j$, and the grid is moved with this velocity; i.e., we add the equations

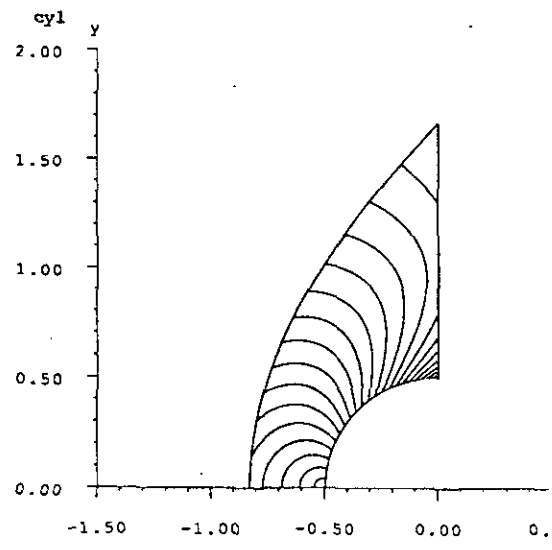

FIG. 6.2. Iso-Mach contours.

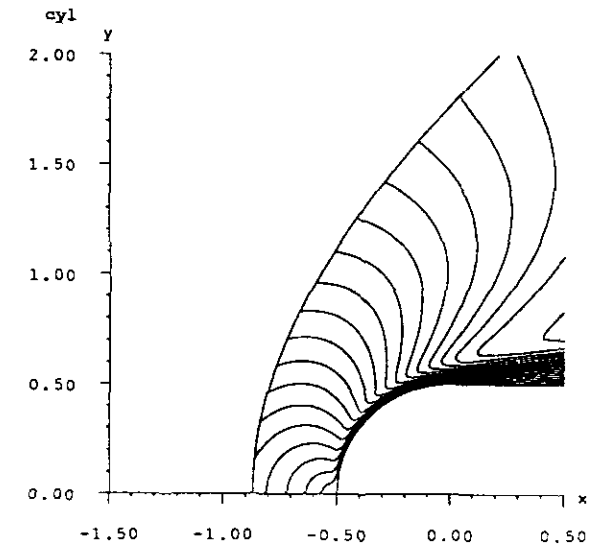

FIG. 6.4. Navier–Stokes solution.

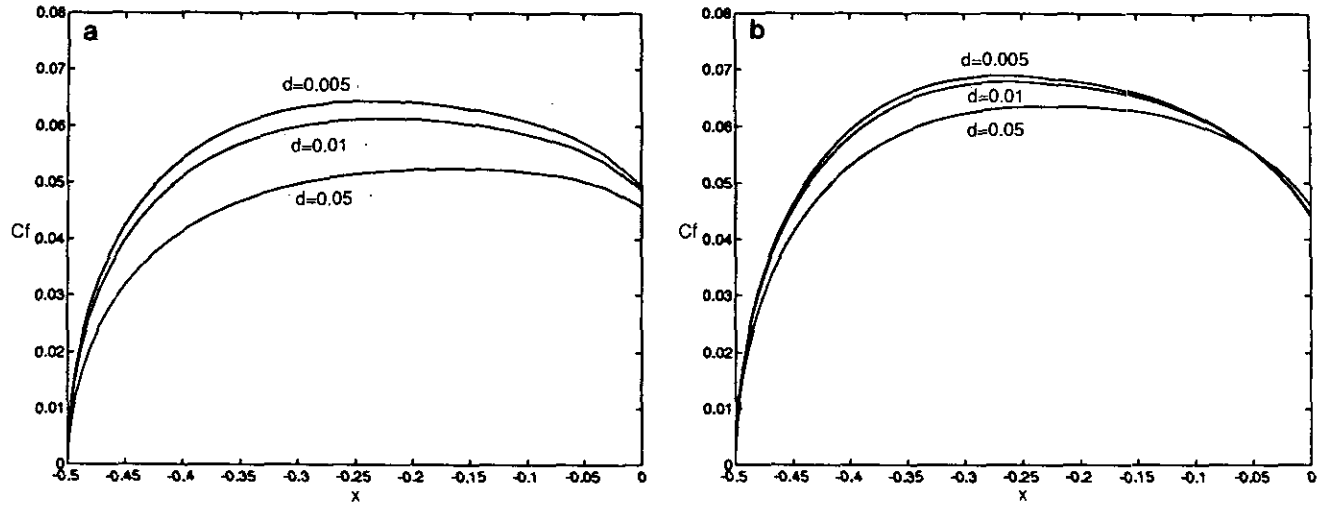

FIG. 6.5. Influence of $\hat{d}$ on C_f for the second order solution: (a) 65 × 33 grid points; (b) 129 × 65 grid points.

$$\frac{dx(i,j,t)}{dt} = sc(i,j)$$

$$\frac{dy(i,j,t)}{dt} = sd(i,j)$$

to the Navier–Stokes system. The grid transformation (x, y) (i, j, t) is thus a function of time. The quantities c and d are two cutoff functions with the properties $d(1, n_j) = c(n_i, n_j) = 0$, $d(n_i, n_j) = c(1, n_j) = 1$, $c(i, 1) = d(i, 1) = 0$. This means that the grid is not moving at the wall and that the outer corners of the grid are only moved one dimensionally along the straight j-coordinate lines.

The metric time derivatives are included in the Navier–Stokes system by taking into account the time dependence of the metric. The system then takes the form

$$(\sqrt{g}\mathbf{w})_t + (y_\eta(\mathbf{f}(\mathbf{w}) - x_t\mathbf{w}) - x_\eta(\mathbf{g}(\mathbf{w}) - y_t\mathbf{w}))_\xi + (-y_\xi(\mathbf{f}(\mathbf{w}) - x_t\mathbf{w}) + x_\xi(\mathbf{g}(\mathbf{w}) - y_t\mathbf{w}))_\eta = 0.$$

We discretize this system in time using a three-stage Runge–Kutta method which we iterate until the time derivatives of the solution and the grid are approximately zero. The Runge–Kutta method, when applied to $u_t = f(u)$, is given by

$$u^{(1)} = u_n + \Delta t f(u_n)$$

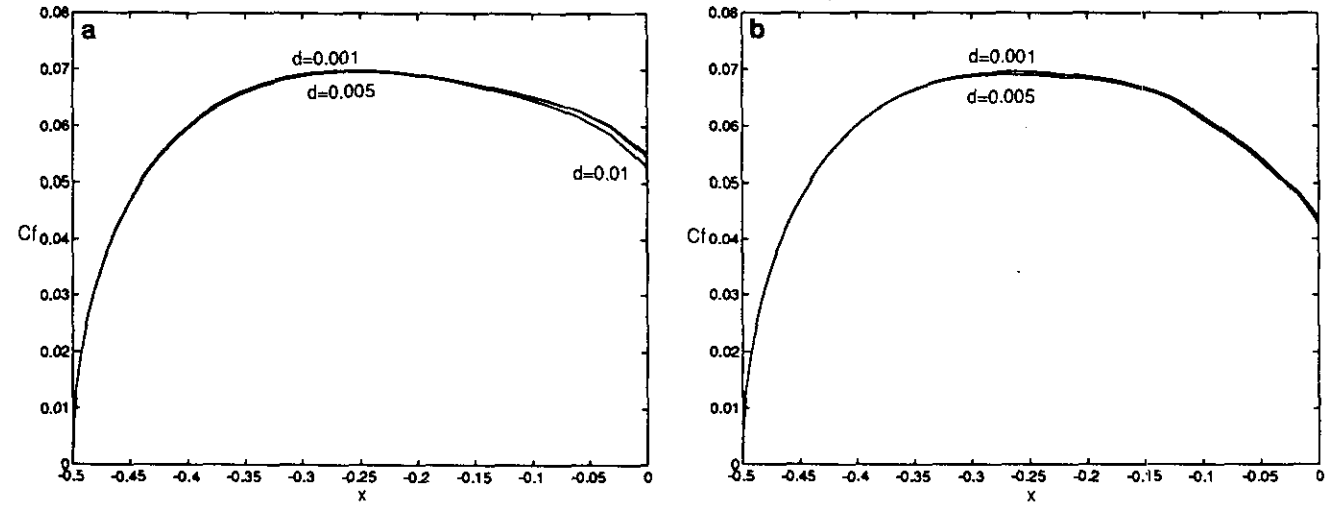

FIG. 6.6. Influence of $\hat{d}$ on C_f for the fourth-order solution: (a) 65 × 33 grid points; (b) 129 × 65 grid points.

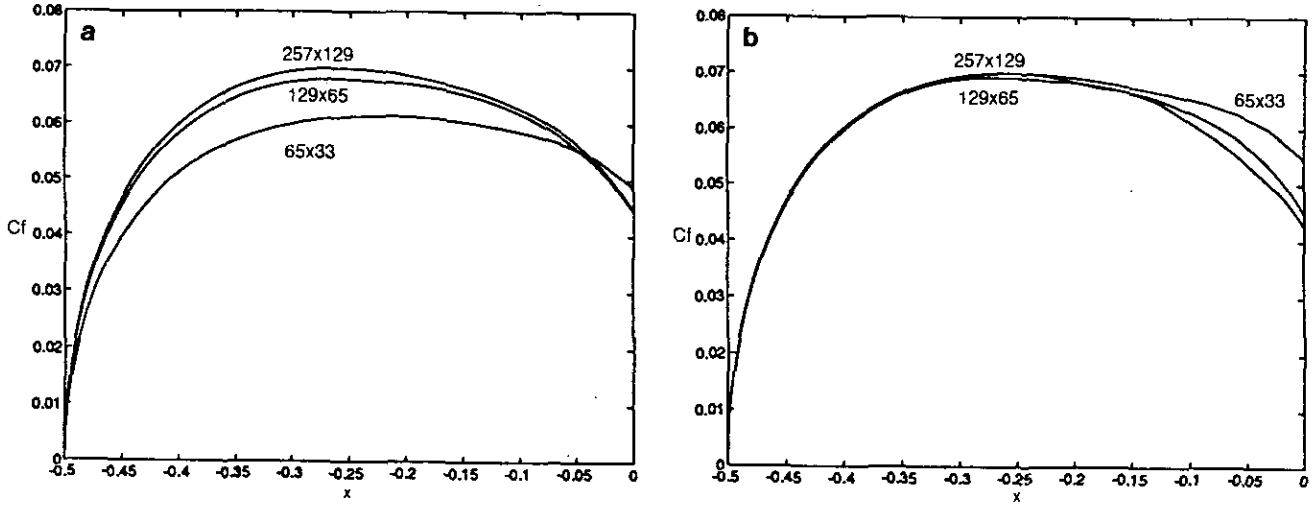

FIG. 6.7. Grid convergence: (a) second order; (b) fourth order.

$$u^{(2)} = u_n + \Delta t((1-\theta)f(u_n) + \theta f(u^{(1)})) \qquad (6.1)$$
$$u_{n+1} = u_n + \Delta t((1-\theta)f(u_n) + \theta f(u^{(2)})),$$

where θ is a parameter, $0 < \theta < 1$. In [6], conditions for Runge–Kutta methods are given which guarantee that if the semi-discrete problem is stable, according to the stability concept used in Section 3, then the time-discretized problem is also stable. It is easy to verify that the method (6.1) satisfies these conditions for $\theta \geq \frac{1}{2}$.

The solution updated in all points, including boundaries, for each stage in the R-K method. Boundary conditions are then imposed. For solid walls, this amounts to setting $u = v = 0$, $T = T_{\text{wall}}$ in the boundary points. For free stream boundaries, we use a characteristic decomposition and give free stream values for ingoing characteristic quantities. The only exception is near the wall, where the free stream boundary cuts off the boundary layer. For the innermost points in such a case, we use a special boundary condition described in [2], which consists of imposing the pressure from outside the boundary layer also inside the boundary layer.

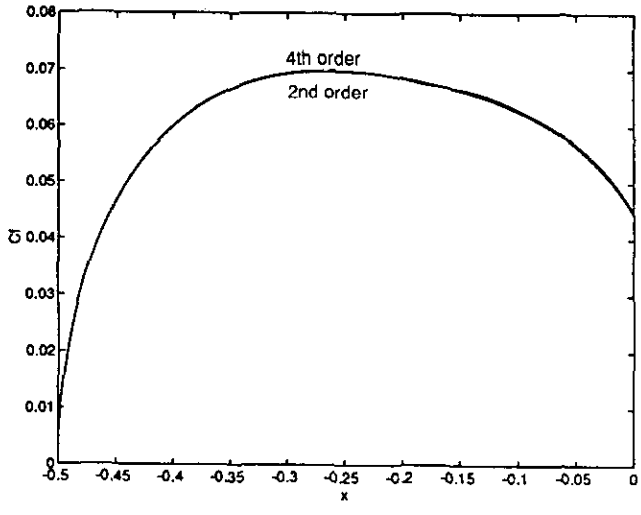

FIG. 6.8. Second- and fourth-order solutions grid converged.

For the cylinder case we first show some Euler solutions, thus with no physical viscosity operator. For this flow we know analytically the entropy on the wall. In Fig. 6.2 we show a typical Euler solution. Table IV shows the L^∞ error in entropy on the wall for a second- and fourth-order method, respectively. Figure 6.3 shows a plot of the 10-logarithm of the error in entropy along the body for the case (D_4, D_{4b}, V_8) on a grid with 65×33 points. All the cases show a similar shape of the error curve, the error is low near the stagnation point and increases exponentially along the body to be largest at the outflow.

Since the dissipation operators have reduced accuracy on the boundary, one could suspect that the convergence rate on the boundary would drop to one and three, respectively. However, the solution converges with a somewhat lower order of convergence. A possible explanation is that the asymptotic region has not been reached for the coarse grids used.

We next go on to the Navier–Stokes equations. An interesting quantity to plot here is the skin friction coefficient, C_f, because it seems to be difficult to capture it correctly. The grid used here is as regular as possible, with aspect ratio of the cells near one, and no stretching towards the wall.

Figure 6.4 shows iso Mach contours of a typical solution, the difference between the fourth- and second-order methods cannot be seen from the contour plots, they look identical.

Instead we present the C_f plots, beginning in Fig. 6.5, where

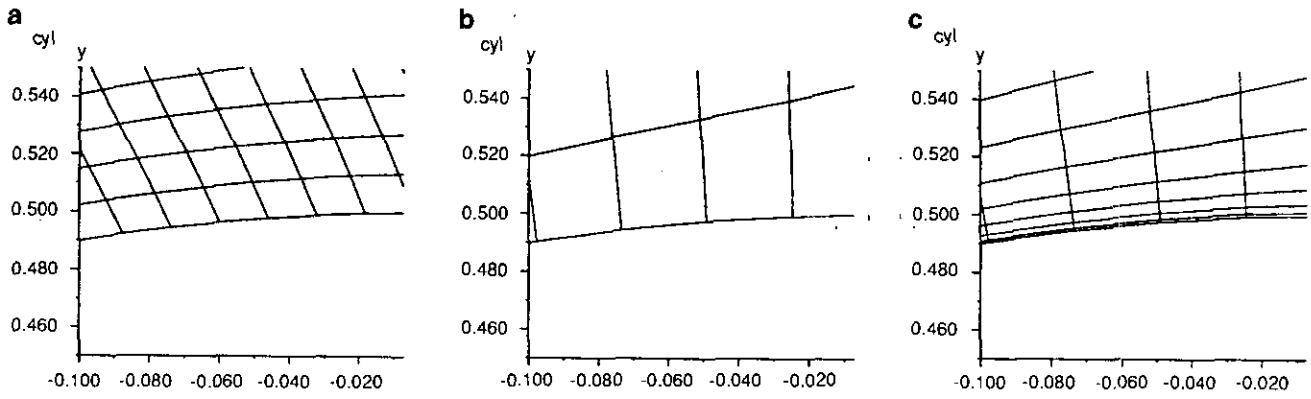

FIG. 6.9. Grid stretchings: (a) finite difference; (b) TVD low stretching; (c) TVD high stretching.

we investigate the influence of the artificial dissipation parameter. For the second-order accurate method, the $\hat{d}$ coefficient of the fourth-order dissipation takes the three values 0.005, 0.01, and 0.05, we see the influence on C_f on a 65×33 grid and a 129×65 grid.

In Fig. 6.6, we present C_f for the fourth-order method, where now $\hat{d}$ is the coefficient of the eighth-order dissipation operator and it takes the values 0.001, 0.005, and 0.01.

We conclude that the eighth-order dissipation operator has not as large effect on C_f as has the fourth-order dissipation, and that in both cases it is possible to choose $\hat{d}$ sufficiently small, so that the influence on the solution is not large.

Next we investigate the grid convergence of C_f for the second- and fourth-order methods. Figure 6.7a shows the second-order solution on three grids with 65×33, 129×65, and 257×29 points, respectively. Figure 6.7b shows the same thing for the fourth-order method. As expected, the fourth-order method converges better, at least in the first half of the flow. Towards the upper part, the flow is more difficult to resolve, probably due to transition to turbulence or separation of the boundary layer.

Figure 6.8 shows a comparison between the finest grid solutions for the second- and fourth-order methods. C_f has converged with grid refinements and to the same curve for both methods of different accuracy. It seems we can trust this solution as being correct.

We next compare this "exact" solution with what one obtains using a conventional scheme used in CFD. We use the second-order accurate TVD, finite volume Osher's method [8], for the Euler part of the equations. The physical viscosity terms are approximated by second-order centered finite volume approximation. The wall boundary conditions are imposed in a standard way, using mirror cells. The solution is iterated to a steady state using a line relaxation iterative method.

The same problem as for the centered finite difference methods is solved, but now with fewer grid points, on a stretched grid. This models a "real" computation. Figure 6.9 shows the difference in stretching between the grids used. With the TVD scheme we capture the shock and thus keep the grid fixed in time.

Figure 6.10 shows iso-mach contours of tvd shock-captured solution, which should be compared with the finite difference solution in Fig. 6.4.

Figure 6.11 shows the C_f curve for the TVD method on a 65×33 grid, for three different stretchings, namely (a) no stretching, (b) stretching as $1/\sqrt{\mathrm{Re}}$, (c) stretching as $1/\mathrm{Re}$. As the stretching increases, the C_f curve seems to converge to a limit curve. The C_f curves on the two most stretched grids are almost indistinguishable.

Finally, the main result of this work, Fig. 6.12 shows the difference between the converged fully resolved finite difference shock-fitted calculation and the highly stretched finite volume shock capturing solution.

As expected, the stretched grid gives a good description of the solution in the region where the boundary layer is well attached to the wall. Towards the upper part of the disk, the

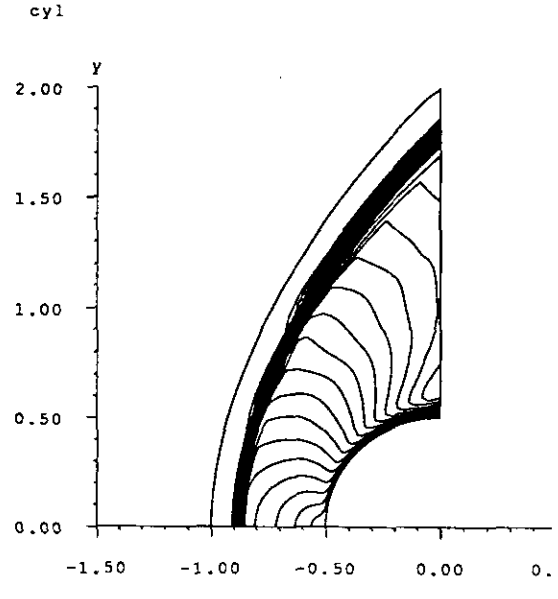

FIG. 6.10. Iso-Mach contours of a TVD solution.

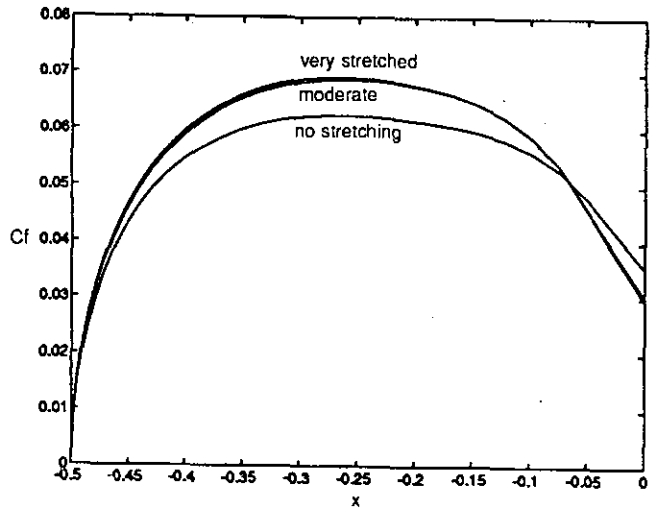

FIG. 6.11. TVD solution, C_f for various stretchings.

boundary layer shows a tendency to separate and the finite volume solution does not give a good description of C_f.

7. CONCLUSION

We have presented two studies. First we have developed stable centered finite difference methods. A stability analysis for a linear scalar test problem was performed. Then, the method was generalized to the compressible two-dimensional Navier–Stokes equations. Some methods of order 8 with very wide stencils were stable for the linear test equations, but were not stable in practice for the 2D Navier–Stokes equations. For other methods of lower accuracy ≤ 4, the stability analysis gave results which agreed with practical computations, in a

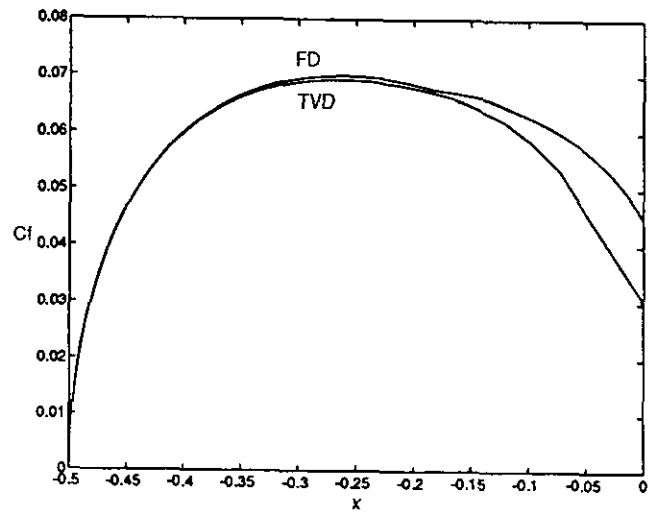

FIG. 6.12 TVD solution on a stretched coarse grid and a converged FD solution.

TABLE V

Interior	Boundary	Inflow, s =	Outflow, s =	Stable
2	2	—	—	Yes
4	2	—	—	Yes
4	3	—	—	Yes
4	4	—	—	Yes
6	2	—	—	Yes
6	3	—	—	Yes
6	4	—	—	Yes
6	5	0.05834 + 1.3373*i*	—	
6	6	0.19633 + 1.3680*i*	—	
8	2	—	—	Yes
8	3	—	—	Yes
8	4	—	—	Yes
8	5	—	—	Yes
8	6	0.15462 + 1.3789*i*	—	
8	7	0.30806 + 1.4135*i*	—	
8	8	0.41745 + 1.4433*i*	—	
10	2	—	—	Yes
10	3	—	0.12006 + 0.95209*i*	
10	4	—	—	Yes
10	5	—	—	Yes
10	6	0.17943 + 1.3401*i*	—	
10	7	0.25946 + 1.4113*i*	—	
10	8	0.39979 + 1.4498*i*	—	
10	9	0.51439 + 1.4748*i*	—	
10	10	0.60554 + 1.4977*i*	—	
12	2	—	—	Yes
12	3	—	0.25660 + 0.96508*i*	
12	4	—	—	Yes
12	5	0.01435 + 1.2745*i*	—	
12	6	0.20389 + 1.3495*i*	—	
12	7	0.23487 + 1.3931*i*	—	
12	8	0.36710 + 1.4420*i*	—	
12	9	0.48822 + 1.4776*i*	—	
12	10	0.59598 + 1.5018*i*	—	
12	11	0.68872 + 1.5200*i*	—	
12	12	0.76663 + 1.5374*i*	—	
14	2	—	—	Yes
14	3	—	0.3359 + 0.9990*i*	
14	4	0.00833 + 1.2486*i*	—	
14	5	0.01389 + 1.2834*i*	0.0491 + 1.3112*i*	
14	6	0.20842 + 1.3647*i*	—	
14	7	0.22599 + 1.3861*i*	—	
14	8	0.35400 + 1.4332*i*	—	
14	9	0.46950 + 1.4708*i*	—	
14	10	0.57543 + 1.5006*i*	—	
14	11	0.67269 + 1.5233*i*	—	
14	12	0.76072 + 1.5403*i*	—	
14	13	0.83886 + 1.5539*i*	—	
14	14	0.90676 + 1.5674*i*	—	

qualitative way. However, exact values about the amount of artificial dissipation needed for stability did not carry over directly to the Navier–Stokes equations.

The second study was to use a fourth order accurate centered difference method, to completely resolve a compressible Navier–Stokes flow past the front part of a disk. This solution was then compared with an unresolved stretched grid TVD shock capturing solution. We concluded that the TVD stretched grid solution gives correct results only if the boundary layer is well attached to the wall.

APPENDIX A

We present in Table V the result from the search for generalized eigenvalues for approximations with pth-order centered differences in the interior of the domain, and rth-order one-sided boundary operators, $r \leq p$. The test equation is the scalar $u_t + au_x = 0$.

The boundary operators are not unique when the order on the boundary is very low. We have here used the centered difference operator of order R, in the boundary points whenever possible. For r odd, we have chosen the operator biased to the left, which means that it is stable for $a > 0$, but not for $a < 0$. This can explain why roots on the outflow appear for some cases with high order in the interior, and with low odd order on the boundary.

REFERENCES

1. B. Fornberg, *Geophysics* **52,** 483 (1987).
2. B. Gustafson and J. Nordström, in *Proceedings, 9th International Conference on Computing Methods in Applied Science and Engineering,* Proceeding Series, (SIAM, Philadelphia, 1990), p. 136.
3. B. Gustafsson, H.-O. Kreiss, and A. Sundström, "Stability Theory of Difference Approximations for Initial Boundary Value Problems II" *Math. Comput.* **26,** 649 (1972).
4. B. Gustafsson, H.-O. Kreiss, and J. Oliger, *Initial-Boundary Value Problems and Its Numerical Solution* (Interscience, New York, 1994).
5. H.-O. Kreiss and J. Oliger, *Tellus* **24,** 199 (1972).
6. H.-O. Kreiss and G. Scherer, *SIAM J. Numer. Anal.* **29,** 640 (1992).
7. P. Olsson, Doctoral thesis, Department of Scientific Computing, Uppsala University, 1992 (unpublished).
8. S. Osher, *SIAM J. Numer. Anal.* **21,** 217 (1984).
9. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, Amsterdam, 1985).
10. M. Thuné, *SIAM J. Sci. Stat. Comput.* **7,** 959 (1986).