JOURNAL OF COMPUTATIONAL PHYSICS 114, 34–44 (1994)

Improved Convergence to the Steady State of the Euler Equations by Enhanced Wave Propagation

PER LÖTSTEDT

Department of Scientific Computing, Uppsala University, Uppsala, Sweden and SAAB-SCANIA, Linköping, Sweden

Received December 7, 1992; revised September 30, 1993

The convergence to the numerical solution of the stationary Euler equations in two and three dimensions is studied. The basic iterative algorithms are Runge–Kutta time-stepping, GMRES, and a modified GMRES method. Convergence acceleration is achieved by two preconditioning techniques: residual smoothing and multigrid iteration. The preconditioners are such that they increase the propagation of smooth error modes out of the computational domain. Runge–Kutta time-stepping and the modified GMRES method guarantee this wave propagation. The results from a number of numerical experiments are reported.

1. INTRODUCTION

The numerical computation of the steady state solution of the equations of compressible fluid flow is a time-consuming process. There is a great interest in the aerospace industry to obtain reliable and fast solutions of these equations. One way of cutting the time that an engineer has to wait for the solution is to buy a faster computer. The other way is to design more efficient numerical algorithms. Here we will discuss and test some improvements of the algorithms for solution of the stationary Euler equations in two and three space dimensions (2D and 3D).

The Euler equations are a system of first-order nonlinear partial differential equations (PDEs). The numerical solution of the PDE problem is often obtained by an iterative method such as Runge–Kutta time-stepping (RK) [16], or the generalized minimum residual method (GMRES) [23]. The convergence of these methods can be explained as a combination of two effects [10, 19, 18]:

- wave propagation of smooth error modes out through the open boundaries,
- damping of the amplitude of oscillatory error modes.

The algorithms we advocate here enhance the wave propagation. Theoretical investigation of linear model problems support this view. Substantial improvements in convergence rate and increased robustness are obtained in numerical experiments. The gain in efficiency of the algorithms is probably also due to better damping of oscillatory modes, but this is more difficult to verify theoretically.

A modified GMRES (mGMRES) method introduced in [11, 12] is tested and compared with the original GMRES and the very successful RK. One advantage of GMRES, mGMRES, and RK in comparison to other iterative schemes, such as the CGS algorithm [9], is that the residual usually has a smooth behavior as the iterations proceed. This property may be crucial for the convergence of strongly nonlinear problems with sign restrictions on the variables. Another advantage is that only residual evaluations are needed in each iterative step. GMRES and mGMRES also have a local optimality property. A disadvantage with GMRES and mGMRES is the extra storage requirements compared to RK. Another problem with GMRES is that the residual for some problems ceases to decrease after a number of iterations. The remedy is to introduce mGMRES or a proper preconditioner. GMRES has been applied to many other flow problems, see, e.g., [28, 3, 4, 26].

We consider two kinds of preconditioning techniques for the Euler equations: residual smoothing and multigrid iteration. Both of them improve the propagation of smooth error waves. The residual smoothing technique in [5] is chosen. It contains the original one [15] as a special case and has better convergence properties and is less sensitive to parameters. Multigrid acceleration has been employed to reduce the computational work in many fluid flow simulations, see, e.g., [22, 14, 28, 15, 20]. The multigrid parameters are selected such that the wave speed of smooth error modes increases.

Section 2 of this paper describes the basic iterative methods RK, GMRES, and mGMRES. In Section 3 the influences of the iterative methods and the preconditioning process on the wave propagation are discussed. The results of the numerical experiments with the Euler equations are reported in Section 4. Finally, conclusions are drawn in Section 5.

0021-9991/94 $6.00

2. BASIC ITERATIVE SCHEMES

Let u denote the vector of m unknown variables and let

$$r(u) = 0 \tag{2.1}$$

be the system of m nonlinear equations obtained after discretization of a first-order PDE such as the Euler equations (see (4.1)). The most popular iterative method to determine u^* satisfying (2.1) for compressible flow problems is Runge–Kutta time-stepping suggested in [16]. One step from n to $n+1$ of the RK(k) algorithm is as follows:

$$\begin{aligned} &u^{(0)} = u^n, \\ &\text{for} \quad j = 1 \text{ to } k \\ &u^{(j)} = u^{(0)} + \alpha_{k+1-j}\, \Delta t\, r(u^{(j-1)}), \\ &u^{n+1} = u^{(k)}. \end{aligned} \tag{2.2}$$

The time-step Δt is proportional to the spatial step h so that $\Delta t\, r(u) = O(1)$ and α_j, $j = 1, \ldots, k$, have some predetermined, fixed values. The residual at step $n+1$ can be written

$$\begin{aligned} &r^{n+1} = r(u^n + \beta_1 r(u^n + \beta_2 r(\cdots u^n + \beta_k r(u^n))\cdots)), \\ &\beta_j = \alpha_j\, \Delta t. \end{aligned} \tag{2.3}$$

Since we want r^n to vanish as quickly as possible an alternative to constant β_j is to let them solve

$$\min_{\beta_j,\, j=1,\ldots,k} \|r^{n+1}\|. \tag{2.4}$$

The norm here and in the sequel is the Euclidean vector norm and its subordinate spectral matrix norm. The problem (2.4) is then solved by the Gauss–Newton method. After linearization the coefficients z_j with definition

$$z_j = \prod_{i=1}^{j} \beta_i \tag{2.5}$$

are given by

$$\min_{z_j,\, j=1,\ldots,k} \left\| r^n + \sum_{j=1}^{k} z_j J^j r^n \right\|, \qquad J = \frac{\partial r}{\partial u}. \tag{2.6}$$

This minimization procedure is the basis for the restarted GMRES(k) algorithm [23] for solution of the linear system

$$\begin{aligned} &J\delta u^n = -r^n, \\ &u^{n+1} = u^n + \delta u^n \\ &\delta u^n \in K_k(J, r^n) = \operatorname{span}\{r^n, Jr^n, \ldots, J^{k-1} r^n\}. \end{aligned} \tag{2.7}$$

The Krylov subspace of dimension k is denoted by K_k. The iteration (2.7) is obtained in [4] by applying Newton's method directly to $r^{n+1} = 0$ and then solving the system of linear equations by GMRES. In [12] the algorithm (2.7) is interpreted as an integration in time of

$$\frac{du}{dt} = \alpha_1 r(u). \tag{2.8}$$

This is obviously also possible for (2.2). Runge–Kutta time-stepping can be implemented by linearizing (2.2) so that

$$u^{n+1} = u^n + \sum_{j=1}^{k} z_j J^{j-1} r^n \tag{2.9}$$

with constants z_j. This is equivalent to solving the system of linear equations in (2.7) by Richardson iteration.

In numerical experiments which deal with linear systems modelling the Euler equations in a channel in [11, 12] the convergence rate of GMRES(k) is sometimes rather poor. A modification of GMRES(k) as introduced in [11, 12] is more successful.

In the modified algorithm, mGMRES(k), the coefficient z_1 in (2.5) is frozen at a value supplied by the user. It follows from the analysis in [19, 12] that z_1 (and α_1 in (2.2) since $z_1 = \alpha_1\, \Delta t$) is responsible for the propagation of smooth error modes. The iterative process for the smooth part approximates the time evolution of the hyperbolic equation (2.8). In the experiments with GMRES(k) in [12] α_1 varies rapidly and is often close to zero resulting in slow convergence. This is avoided in mGMRES(k). The remaining $k-1$ coefficients z_i are chosen to minimize the linearization of r^{n+1} in (2.6).

One step of mGMRES(k) is

1. $r^n = r(u^n)$, $\beta = \|r^n\|$, $c_1 = 1/\beta$, $v_1 = r^n c_1$,
2. for $j = 1$ to k

$$\begin{aligned} &\text{for } i = 1 \text{ to } j \quad h_{ij} = v_i^T J v_j, \\ &\hat{v}_{j+1} = Jv_j - \sum_{i=1}^{j} h_{ij} v_i, \\ &h_{j+1,j} = \|\hat{v}_{j+1}\|, \\ &v_{j+1} = \hat{v}_{j+1}/h_{j+1,j}, \\ &c_{j+1} = -h_{j+1,j}^{-1} \sum_{i=1}^{j} c_i h_{ij}, \\ &\text{end for } j \end{aligned} \tag{2.10}$$

3. Solve

$$\min_{y,\, z_1 = c^T y} \|H_k y - \beta e_1\|,$$

4. $u^{n+1} = u^n + V_k y$.

In the algorithm e_1 denotes the first unit vector $(1, 0, ..., 0)^T$ and

$$H_k \in R^{(k+1)\times k}, \qquad (H_k)_{ij} = h_{ij},$$
$$V_k \in R^{m\times k}, \qquad V_k = (v_1, v_2, ..., v_k),$$
$$c \in R^k.$$

In step 2 an orthonormal basis of $K_k(J, r^n)$ is generated by Gram–Schmidt orthogonalization as suggested in [23]. The minimization problem in step 3 is solved as in [12] by first eliminating the linear constraint and then computing y from the reduced system in a standard fashion [1, p. 592]. There is a linear relation between z in (2.5) and y in (2.10)

$$z = Cy \tag{2.11}$$

derived in [21]. In (2.10) we use only the first row c of C.

The GMRES(k) algorithm differs from (2.10) in steps 2 and 3. There is no need to update c in the j-loop and the linear least squares problem to be solved has no linear equality constraint.

mGMRES(k) has good wave propagation properties while GMRES(k) may exhibit poor convergence behavior for smooth error modes. Both methods determine the free coefficients z_i (or α_i) adaptively. For problems where the wave transport is unimportant to the convergence rate GMRES(k) is expected to be the preferred method since it has one more coefficient to vary.

The vector Jv_j in (2.10) is approximated as in [4] by

$$Jv_j = \sigma^{-1}(r(u^n + \sigma v_j) - r(u^n)), \tag{2.12}$$

where σ is sufficiently small. The Jacobian J is never calculated explicitly. The most expensive part in (2.10) to compute is the residual r in step 1 and (2.12). In (2.10) $k+1$ such computations are necessary. This number can be reduced to k for linear problems if r^n is updated as described in [23]. It may be possible to use the same idea for nonlinear problems, but with occasional full evaluations of r^n in step 1.

The number of floating point operations to calculate r is of the order of 100 per component of r. Let m be the number of unknown variables. Then the number of operations in GMRES(k) is found in [23] to be about $2mk^2$ plus the k computations of Jv_i. Slightly more operations are required by mGMRES(k).

Since

$$100m(k+1) \gg 2mk^2,$$

the computational cost is dominated by the evaluation of r. This conclusion is confirmed in the numerical examples in Section 4, where the time per iteration using (m)GMRES(3) is about $\frac{4}{3}$ of the time of RK(3).

The RK(k) algorithm (2.2) only needs k computations of r and a total of about $100mk$ operations.

When m is very large, $m \gg k$, the storage requirements of algorithm (2.10) and (2.12) are approximately $m(k+3)$ in a straightforward implementation. This is a disadvantage in comparison to RK(k) in (2.2), where space for only three m-vectors is needed in the primary memory independent of k. Therefore, k must be kept small in (2.10). The convergence to the steady state solution is slower for small k [12], but the risk of stagnation because of lost orthogonality between the v_i-vectors due to the round-off errors is reduced [27, 17]. Problems with slow convergence are solved better by the introduction of an efficient preconditioner than increasing k.

The reason why mGMRES(k) represents an improvement of GMRES(k) for first-order PDEs is that wave propagation properties are guaranteed in mGMRES(k) by $z_1 \neq 0$. This is analyzed in [11, 12] and is discussed further in the next section. In [7] a combination of GMRES(k) and mGMRES(k) is tested by letting y in step 3 satisfy an inequality constraint

$$c^T y \geqslant z_1.$$

3. PRECONDITIONING FOR IMPROVED WAVE PROPAGATION

It is well known that the efficiency of RK(k) and GMRES(k) can improve dramatically by preconditioning of the problem (2.1). Here we will discuss two techniques that enhance the propagation of smooth error modes out through the open boundaries of the computational domain: residual smoothing and multigrid iteration. The implicit–explicit residual smoothing algorithm described here is developed in [5, 6] and extended to 3D and tested for simple 3D problems in [25]. The multigrid scheme is a V-cycle, see [2; 13, Section 4.1], with extra smoothing iterations on the coarse grids. We show in this section by Fourier analysis how the wave speed of smooth modes is increased by these techniques.

In order to understand the effects of the preconditioning, let us consider the model problem

$$au_x + bu_y = f \tag{3.1}$$

in the (x, y)-plane. Discretize (3.1) on a grid G,

$$G = \{(\mu h, \nu h) \mid \mu, \nu \in Z\},$$
$$Z = \{\text{the integer numbers}\}, \tag{3.2}$$

by, e.g., the second-order accurate centered difference

approximation. Let f and the initial guess u^0 have compact support and iterate with RK(k) (2.2), GMRES(k), or mGMRES(k) (2.10). Then it suffices to study (3.1) with $f=0$ and let u^n be the iteration error after n steps.

As in [19, 12] the error u^n can be written on G in (3.2) using its Fourier representation

$$\begin{aligned} u^n_{\mu\nu} &= \int_D \exp(i(\xi_1\mu+\xi_2\nu)h)\,\hat{u}^n(\xi)\,d\xi, \\ \xi &= (\xi_1,\xi_2)^{\mathrm{T}}, \\ D &= [-\pi/h,\pi/h]\times[-\pi/h,\pi/h]. \end{aligned} \tag{3.3}$$

The smooth part of $u^n_{\mu\nu}$ is defined by the low wave numbers ξ:

$$\begin{aligned} (u^n_{\mu\nu})_S &= \int_{D_0} \exp(i(\xi_1\mu+\xi_2\nu)h)\,\hat{u}^n(\xi)\,d\xi, \\ D_0 &= \{\xi_i,\ i=1,2 \mid |\xi_i|\leqslant\xi_{\max}\}, \qquad \xi_{\max}=O(1). \end{aligned}$$

Let the time-step Δt in (2.3) be

$$\Delta t=\kappa h, \qquad \kappa=O(1). \tag{3.4}$$

Then it is shown in [12] that with RK(k), GMRES(k), or mGMRES(k) and α_1 defined by (2.3) and (2.5)

$$\begin{aligned} (u^{n+1}_{\mu\nu})_S &= \int_{D_0} \exp(i(\xi_1\mu+\xi_2\nu)h)\,\hat{u}^{n+1}(\xi)\,d\xi \\ &\approx \int_{D_0} \exp(i(\xi_1(\mu+\alpha_1\kappa a) \\ &\quad +\xi_2(\nu+\alpha_1\kappa b))h)\,\hat{u}^n(\xi)\,d\xi \\ &= (u^n_{\mu+\alpha_1\kappa a,\nu+\alpha_1\kappa b})_S. \end{aligned} \tag{3.5}$$

The interpretation of (3.5) is that the smooth part of u at step n has been propagated on the grid by $-\alpha_1\kappa a$ in the x-direction and $-\alpha_1\kappa b$ in the y-direction. The distance that the wave moves on G per step depends on α_1 in the iterative method, the κ parameter in (3.4), and the coefficients a and b in the differential equation (3.1).

After a sufficient number of steps the smooth error modes have been expelled from a finite computational domain by wave propagation, provided that the numerical boundary conditions allow outgoing waves to pass without reflection. The smooth error modes are not damped very well by Krylov subspace methods. In RK(k) and mGMRES(k) α_1 is a constant parameter different from zero. In GMRES(k) the wave propagation property of smooth modes is not always present, since α_1 may tend to zero.

The multigrid V-cycle is analyzed with the same Fourier technique as above in [19, 18]. The iterative scheme used as a smoother is invoked on all grids, also the coarsest. This is common practice in compressible flow computations. Let p and q denote the number of presmoothing and postsmoothing iterations, respectively, and let l be the number of grids in the V-cycle. The step-size h is doubled when descending from one grid to the next coarser grid. Then it is shown in [18] for RK(k) that under certain assumptions

$$(u^{n+1}_{\mu\nu})_S \approx (u^n_{\mu+\alpha_1\kappa\chi a,\nu+\alpha_1\kappa\chi b})_S,$$

where

$$\chi=(p+q)(2^l-1). \tag{3.6}$$

Even for a small number of grids, a substantial increase of the wave speed by a factor χ is the result of one V-cycle. Multigrid iteration with GMRES(k) as the smoother is more difficult to analyze, but numerical experiments in [18] indicate a behavior similar to (3.6).

With the implicit–explicit residual smoothing algorithm in 2D the preconditioned $\tilde{r}^n$ is obtained as

$$\tilde{r}^n=\alpha T_y^{-1}T_x^{-1}Sr^n. \tag{3.7}$$

This $\tilde{r}^n$ then replaces r^n in the iterative schemes in Section 2. The tridiagonal matrices T_x and T_y are such that

$$u_i=(T_w v)_i=(1-\beta\delta_w^2)\,v_i, \qquad w=x \text{ or } y,$$

where δ_w^2 is the second difference in the w-direction. Since T_w is tridiagonal it is easily inverted in (3.7). In 2D β is chosen to be

$$\beta=(\alpha^2-1)/4.$$

The matrix S in (3.7) is defined by

$$u_i=(Sv)_i=(1-\gamma(\delta_x^2+\delta_y^2)+\delta\delta_x^2\delta_y^2)\,v_i.$$

The parameters δ and γ are

$$\begin{aligned} \gamma &= (\alpha-1)/4, \\ \delta &= (\alpha^3-8\gamma-1)/16. \end{aligned}$$

The boundary conditions for δ_w^2 are of Dirichlet type. In the implicit smoother in [15] $\delta=\gamma=0$.

In the 3D version the right-hand side of (3.7) is premultiplied by T_z^{-1} and additional terms are needed in S to balance T_z^{-1}. Then S will be such that

$$\begin{aligned} u_i=(Sv)_i=(1-\gamma(\delta_x^2+\delta_y^2+\delta_z^2) \\ +\delta(\delta_x^2\delta_y^2+\delta_x^2\delta_z^2+\delta_y^2\delta_z^2)-\varepsilon\delta_x^2\delta_y^2\delta_z^2)\,v_i. \end{aligned}$$

The parameters are

$$\beta = (\alpha^2 - 1)/4,$$
$$\gamma = (\alpha - 1)/4,$$
$$\delta = (\alpha^3 - 8\gamma - 1)/16,$$
$$\varepsilon = (\alpha^5 - 48\delta - 12\gamma - 1)/64.$$

For a detailed discussion of the algorithm see [5, 6]. Apply the implicit–explicit residual smoother (3.7) to the model problem (3.1) with $f = 0$ and RK(k), GMRES(k), or mGMRES(k) as the iterative method. Then using the Fourier representation (3.3) one can show that

$$(u_{\mu\nu}^{n+1})_S \approx (u_{\mu+\alpha_1\kappa\alpha a,\nu+\alpha_1\kappa\alpha b}^{n})_S. \tag{3.8}$$

The preconditioning (3.7) increases the speed of the smooth waves by the factor α (cf. (3.5) and (3.6)). The choice of parameters β, δ, and γ (β, γ, δ and ε) in 2D (3D) is such that $\tilde{r}^n$ in (3.7) is close to r^n for large wave numbers ξ. For small $\|\xi\|$ both the wave propagation and the damping properties are improved in the iterative process.

A well-known and simple preconditioner is "local time-stepping." The idea is to let Δt be as large as possible in each computational cell based on a local stability criterion. The result is a diagonal scaling of the residual,

$$\tilde{r}^n = Dr^n,$$
$$D = \operatorname{diag}\{\kappa_1, \kappa_2, \ldots, \kappa_m\}, \tag{3.9}$$

where m is the number of equations, cf. κ in (3.4). This can be regarded as an attempt to make the maximum wave speed equal in all cells.

We have found in this section that for a model problem (3.1) RK(k) and mGMRES(k) have good wave propagation properties. Moreover, the multigrid V-cycle and the residual smoothing algorithm increase the wave speed.

4. NUMERICAL EXAMPLES

The convergence of the iterative algorithms in Sections 2 and 3 to the solution of the stationary Euler equations is studied in numerical experiments in this section. In 2D the geometries are two channels and a wing profile. In 3D the solution is computed in a channel and around a wing. The number of iterations and the total CPU-time are compared for RK(3), GMRES(3), and mGMRES(3) combined with different preconditioners. The number of stages for the GMRES-methods must be low in order to reduce the storage requirements. There is a reasonably efficient RK-algorithm with three stages; see (4.4). For these reasons and for a fair comparison we have chosen to let all the methods have three stages and therefore three acceleration parameters α_i. There exist RK(5) methods that sometimes are more efficient then RK(3) but also (m)GMRES(5) often beats (m)GMRES(3) [12]. The comparisons are made in the same environment and the relative difference between the methods is of greater interest than the absolute measures of their performance.

The stationary Euler equations of compressible fluid flow are in 3D:

$$(\rho u)_x + (\rho v)_y + (\rho w)_z = 0,$$
$$(\rho u^2 + p)_x + (\rho uv)_y + (\rho uw)_z = 0,$$
$$(\rho vu)_x + (\rho v^2 + p)_y + (\rho vw)_z = 0,$$
$$(\rho wu)_x + (\rho wv)_y + (\rho w^2 + p)_z = 0,$$
$$(\rho Eu + up)_x + (\rho Ev + vp)_y + (\rho Ew + wp)_z = 0.$$

To close the system, we need the perfect gas law for the pressure

$$p = (\gamma - 1)\,\rho(E - (u^2 + v^2 + w^2)/2), \tag{4.1}$$

where $\gamma = 1.4$.

The density is ρ, E is the total energy, p is the pressure, and the velocity vector is $(u, v, w)^T$. The 2D version is obtained by letting $w \equiv 0$ and $\partial(\cdot)/\partial z \equiv 0$ in (4.1).

The basis for this investigation is the computer code described in [24] for numerical solution of the stationary Euler equations in 2D and a code developed from an early FLO-program by A. Jameson for the corresponding 3D problem. The single grid RK time-stepping part of the 3D program is essentially the original FLO-implementation. The discretization in 2D and 3D is achieved as in [16, 15], i.e., a finite volume approximation of the first derivatives leading to a second order centered difference approximation on Cartesian grids with constant step size and a blend of second and fourth differences as artificial viscosity. The latter part of the artificial viscosity can be written for the model equation (3.1),

$$\frac{\theta}{64}\frac{\lambda}{h}(\delta_x^4 u + \delta_y^4 u), \tag{4.2}$$

where λ is proportional to $\|(a, b)\|$ and θ is a parameter. The numerical boundary conditions in the far field are based on Riemann invariants to permit outgoing modes to cross the outer artificial boundary without severe reflections. This spatial discretization defines $r(u)$ in (2.1).

The implicit–explicit residual smoothing is implemented as described in Section 3. The multigrid algorithm is the FAS scheme [2; 13, Section 9.3]. In the cell-centered finite volume discretization one cell on a coarse grid corresponds

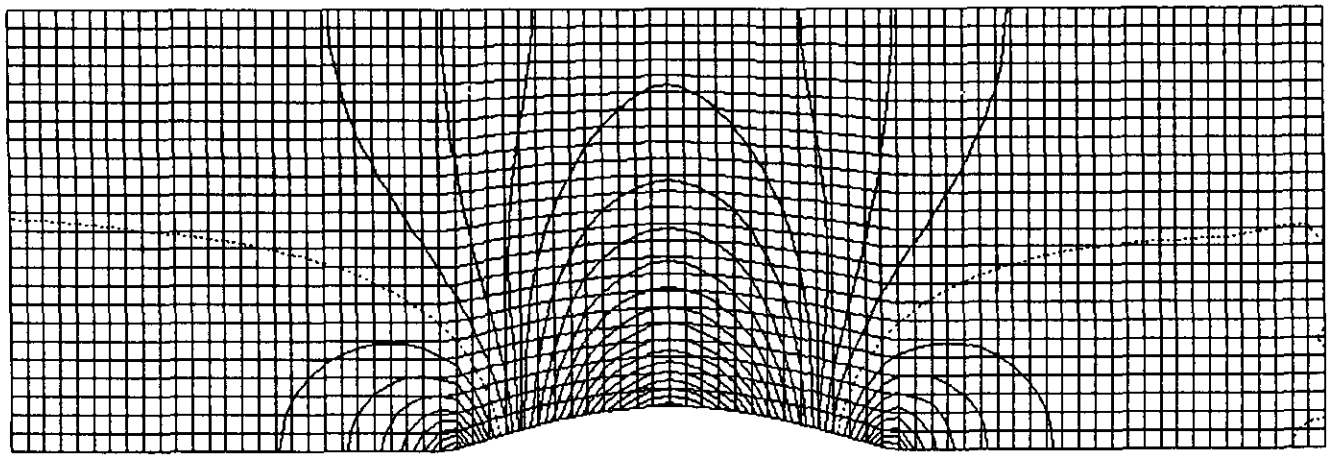

FIG. 1. The grid of the 2D channel with a bump. The isobars of the solution with $M_\infty = 0.5$ are plotted.

to 4 (8) cells on the next finer grid in 2D (3D). The restriction from a fine to a coarse grid is therefore determined in 2D (3D) as an area (volume) weighted average of the 4 (8) values of the cells on the fine grid that combine to one cell on the coarse grid. The prolongation back from the coarse to the fine grid is obtained by bilinear (trilinear) interpolation in 2D (3D). Care is taken to ensure that the smoothness of the interpolated corrections is preserved at the boundaries of the computational domain.

The RK(k), GMRES(k), and mGMRES(k) methods are implemented as in (2.2) and (2.8). The termination criterion is

$$\left(m^{-1}\sum_{i=1}^{m}(r_i^n)^2\right)^{1/2} = \|r^n\|/\sqrt{m} \leqslant \varepsilon. \tag{4.3}$$

Usually only the residual of the first equation of (4.1) is estimated, but (4.3) is more reliable and suitable for GMRES.

There is a conflict between the accuracy of the converged solution and the number of iterations necessary to obtain that solution. Given the method of discretization of our equations and the treatment of the numerical boundary conditions, the accuracy is affected by the computational grid, the artificial viscosity (4.2), and the termination criterion (4.3). The truncation error in the approximation of the first derivative terms in (4.1) decreases with finer grids. Finer grids often imply slower convergence. A small θ in (4.2) improves the accuracy but slows down the convergence. Fewer iterations are needed if ε in (4.3) is increased but then the accuracy deteriorates. In our numerical examples we have chosen to keep the grid fixed and then determine appropriate values for θ in (4.2) and ε in (4.3) experimentally so that their influence on the converged solution is small. In most of our computations $\varepsilon = 1.0\text{E}-3$.

The variables in (4.1) and a characteristic length are scaled so that they all are in the neighborhood of 1. Hence, r^n in (4.3) is non-dimensionalized. Then following the recommendation in [4], σ in (2.12) is taken to be a constant ($\sigma = 1.0\text{E}-6$ in our case).

TABLE I

Convergence Comparison for the 2D Subsonic Channel with a Bump

	RK(3)		GMRES(3)		mGMRES(3)	
	No. of iter.	CPUs	No. of iter.	CPUs	No. of iter.	CPUs
B	4712	144.39	S	S	1294	57.83
RS	839	31.81	S	S	347	19.07
MG1	280	26.82	263	30.38	180	20.88
MG2	98	22.09	50	15.41	61	18.68

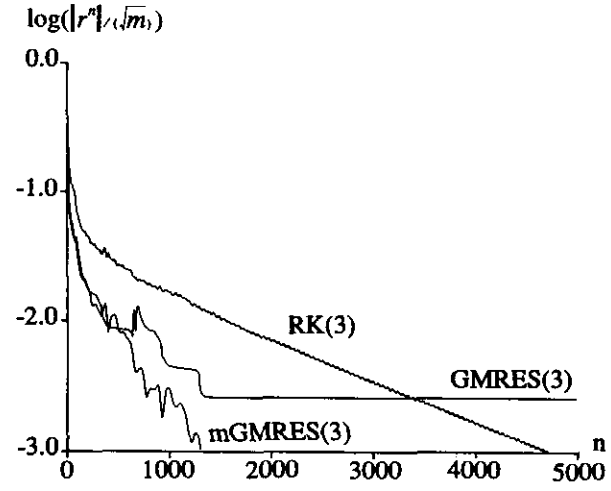

FIG. 2. A comparison of the convergence histories of RK(3), GMRES(3), and mGMRES(3) for the subsonic flow in the 2D channel.

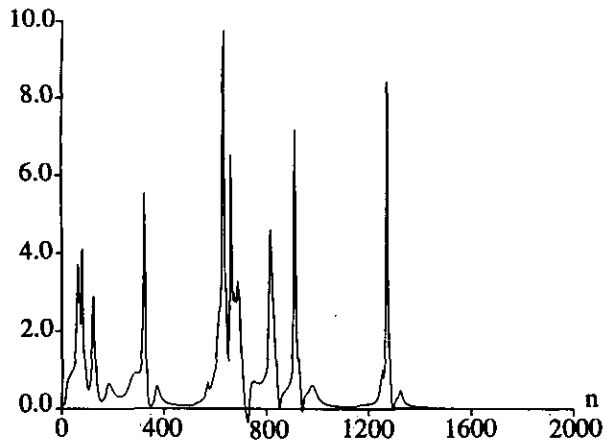

FIG. 3. The α_1 parameter of GMRES(3) for the subsonic flow in the 2D channel.

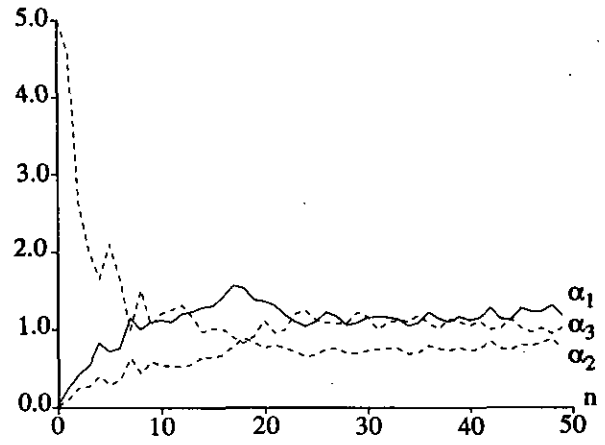

FIG. 5. The α_i parameters of GMRES(3) with residual smoothing and multigrid iteration for the subsonic flow in the 2D channel.

The parameters α_i, $i = 1, 2, 3$, in RK(3) in (2.2) are

$$\alpha_1 = 1, \qquad \alpha_2 = 0.6, \qquad \alpha_3 = 0.6. \tag{4.4}$$

The z_1-parameter in mGMRES(3) in (2.10) is (cf. (2.3), (2.5))

$$z_1 = \alpha_1 \Delta t = \Delta t. \tag{4.5}$$

The number of iterations and the CPU-time in seconds on one processor of CRAY X-MP are compared for three convergence acceleration algorithms with RK(3), GMRES(3), and mGMRES(3) as basic iterative methods. The basis for the comparison is iteration with only the basic iterative method (B). Then the basic method is combined with the preconditioning developed in [5] with $\alpha = 3$, see Section 3 (RS). In numerical experiments [5, 25] the method (3.7) is not very sensitive to the choice of α. The second convergence accelerator is the multigrid V-cycle on three grids ($l = 3$) with one presmoothing iteration ($p = 1$) on each grid (MG1). Only a second difference artificial viscosity is applied on the coarse grids.

Finally, the preconditioning technique in [5] with $\alpha = 3$ is combined with a multigrid V-cycle on three grids with extra iterations on the coarse grids (MG2). On the coarsest grid $p + q = 10$, on the intermediate grid $p = q = 3$ and on the finest grid $p = 1$, $q = 0$. The cost of extra iterations on coarse grids is small, expecially in 3D, and has a positive influence on the convergence rate. The approach is analyzed and tested on two grids in [8]. Preconditioning by "local time-stepping" is used in all four cases.

The first example is a 2D channel with solid upper and lower walls and a bump. In the first four examples the flow enters the computational domain from the left and leaves to the right. The Mach number M_∞ is 0.5, the number of cells is 80×24 and θ in (4.2) is 0.5. The discretization and isobar lines are plotted in Fig. 1. The comparison between the different algorithms is found in Table I. In the table S denotes stagnation of the iteration; i.e., there is no apparent progress towards a converged solution. mGMRES(3) is here faster than RK(3) in all rows of the table. The best

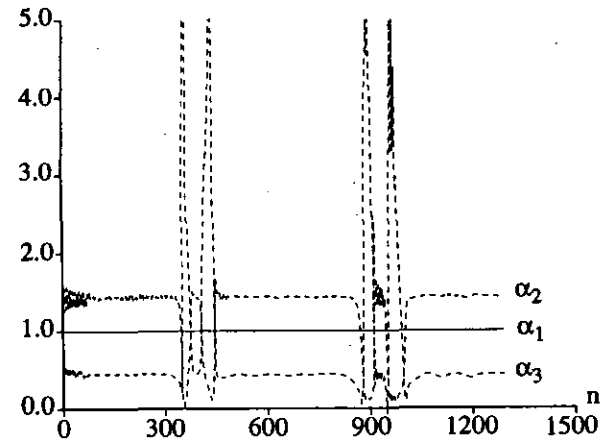

FIG. 4. The α_i parameters of mGMRES(3) for the subsonic flow in the 2D channel.

TABLE II

Convergence Comparison for the 2D Supersonic Channel with a Ramp

	RK(3)		GMRES(3)		mGMRES(3)	
	No. of iter.	CPUs	No. of iter.	CPUs	No. of iter.	CPUs
B	713	43.32	S	S	725	65.77
RS	292	23.27	S	S	299	34.40
MG1	173	27.57	241	44.44	147	29.93
MG2	157	54.26	130	63.57	114	55.59

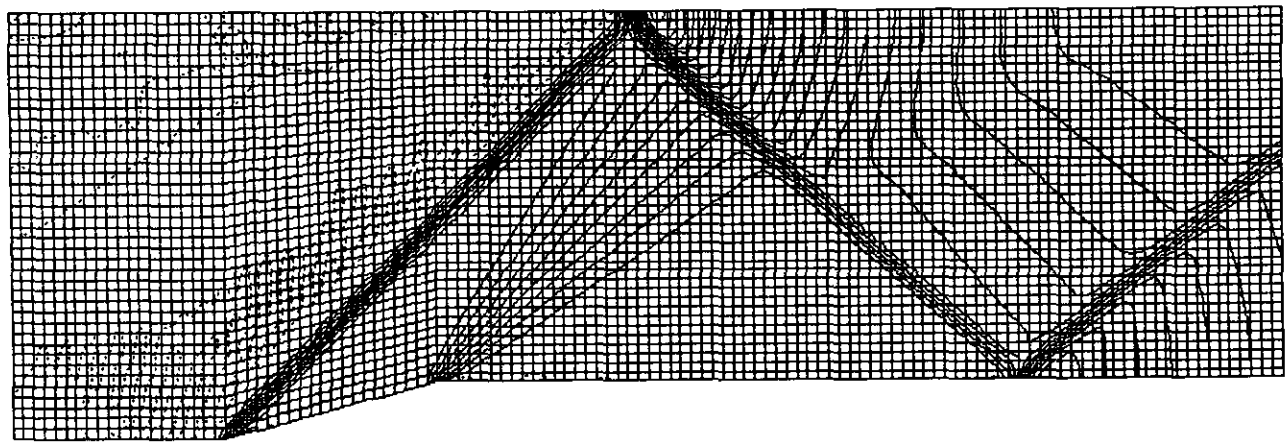

FIG. 6. The grid of the 2D channel with a ramp. The isobars of the solution with $M_\infty = 2.0$ are plotted.

result is obtained by GMRES(3), combined with residual smoothing and multigrid iteration with extra coarse grid smoothing (MG2). The stagnation of GMRES(3) without preconditioning is illustrated in Fig. 2, where the convergence history of $\|r^n\|/\sqrt{m}$ is plotted as a function of n. For comparison the convergence of RK(3) and mGMRES(3) is included. The three coefficients in y in (2.10) can be transformed to the Runge–Kutta parameters α_i, $i = 1, 2, 3$, in (2.3) via (2.11) and (2.5). Then the variation of the leading parameter α_1 of GMRES(3) of importance to the wave propagation is depicted in Fig. 3. After 1400 iterations α_1 approaches zero and α_2 and α_3 start oscillating wildly. This is the point where the stagnation phase of GMRES(3) begins in Fig. 2. For comparison the α-coefficients of mGMRES(3) are plotted in Fig. 4. In Fig. 5 the coefficients of GMRES(3) on the fine grid in combination with MG2 are shown. In both cases α_i has a relatively smooth behavior. The convergence histories of RK(3) in (2.2) and (2.9) are indistinguishable in a plot. Both algorithms need 4712 iterations to converge.

The grid and the isobar solution of the next 2D example is displayed in Fig. 6. The inflow speed M_∞ is 2.0 and a shock is generated by the foot of the ramp. The number of cells is 120×40 and θ in (4.2) is 0.5. The results are presented in Table II. Here the multigrid algorithm (MG1, MG2) reduces the number of iterations but not the CPU-time. This is probably due to the presence of the shocks, which are not treated properly on the coarsest grid. The best alternative is here RK(3) and residual smoothing (RS). Note that the number of iterations of RK(3) and

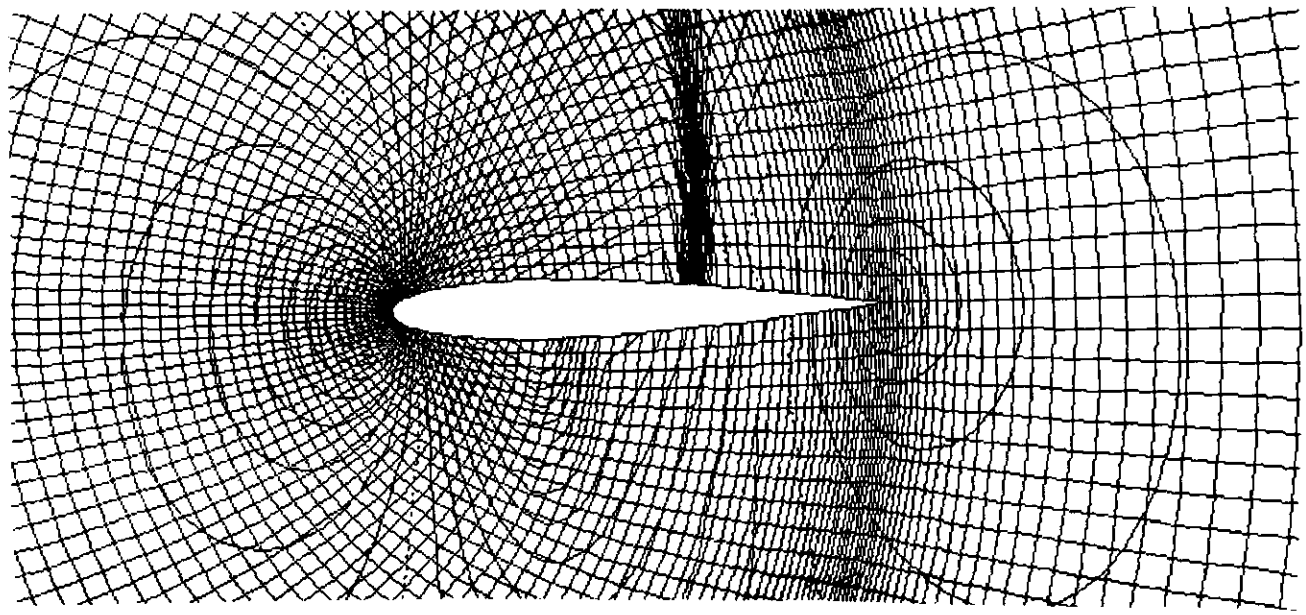

FIG. 7. The inner part of the grid around NACA0012. The isobars of the solution with $M_\infty = 0.8$ and $\alpha = 1.25°$ are plotted.

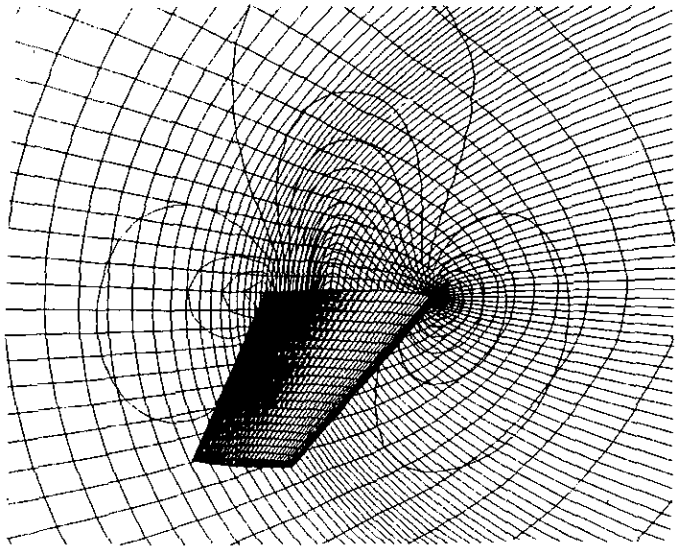

FIG. 8. The grid on the M6 wing and the symmetry plane. The isobars of the solution with $M_\infty = 0.84$ and $\alpha = 3.06°$ are plotted on the symmetry plane.

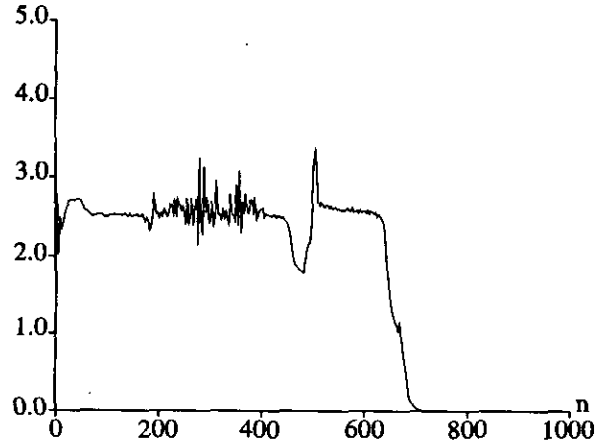

FIG. 9. The α_1 parameter of GMRES(3) for the transonic flow around the M6 wing.

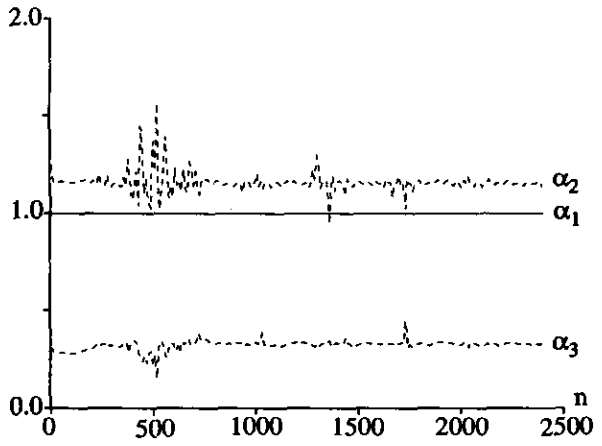

FIG. 10. The α_i parameters of mGRES(3) for the transonic flow around the M6 wing.

TABLE III

Convergence Comparison for the Transonic Airfoil

	RK(3)		GMRES(3)		mGMRES(3)	
	No. of iter.	CPUs	No. of iter.	CPUs	No. of iter.	CPUs
B	1998	266.81	S	S	2753	548.12
RS	1212	224.85	1294	336.71	1088	289.87
MG1	542	156.41	401	144.35	401	150.14
MG2	82	58.40	62	60.66	81	76.27

mGMRES(3) in MG1 and MG2 is almost the same but RK(3) is cheaper per iteration. With the combination RK(3) and MG2 the local time-step had to be lowered to obtain convergence. A diagram of α_1 as a function of n for GMRES(3) looks similar to Fig. 3. After stagnation ($n \approx 400$) α_1 is close to zero and α_2 and α_3 oscillate. The α_i-coefficients of mGMRES(3) are much smoother.

The next example is a standard 2D test case for transonic flow around a NACA0012 airfoil. The angle of attack α is 1.25°, $M_\infty = 0.8$, $\theta = 1.0$, and the grid has 193×61 cells. The inner part of the grid and isobars of the solution are depicted in Fig. 7. The comparisons between different algorithms are made in Table III. When comparing the number of iterations RK(3) is not the best basic iterative scheme, but because the work per iteration is smaller the total CPU-time is often the smallest. For this example ε in (4.3) was changed from 1.0E − 3 to 1.0E − 6. The number of iterations for RK(3) and MG2 increased to 137 but there was no visible difference in the converged solutions. With $\theta = 0.5$ and $\theta = 2.0$ the number of iterations needed for $\varepsilon = 1.0\text{E} - 3$ was 117 and 69, respectively. This result indicates that the damping of oscillatory error modes also is of importance to the convergence rate. The difference between the solutions on the edge of the airfoil was noticeable only at the shocks which moved about 0.5% of the chord.

The first 3D example is the same channel as in Fig. 1 extended by 16 cells in the direction normal to the grid in Fig. 1. Thus, the number of cells is $80 \times 24 \times 16$. The

TABLE IV

Convergence Comparison for the 3D Subsonic Channel with a Bump

	RK(3)		GMRES(3)		mGMRES(3)	
	No. of iter.	CPUs	No. of iter.	CPUs	No. of iter.	CPUs
B	3504	2148.93	S	S	1413	1273.20
RS	1077	1013.81	S	S	452	591.86
MG1	328	402.33	2212	3238.28	164	265.24
MG2	76	200.45	33	124.37	52	190.15

TABLE V

Convergence Comparison for the M6 Wing

	RK(3)		GMRES(3)		mGMRES(3)	
	No. of iter.	CPUs	No. of iter.	CPUs	No. of iter.	CPUs
B	2429	2758.60	S	S	2418	3849.19
RS	901	1566.23	697	1750.54	889	2045.26
MG1	S	S	140	423.88	327	892.29
MG2	59	289.90	54	354.93	60	403.45

parameter θ is 0.5 and $M_\infty = 0.5$. As in Table I GMRES(3) with the most elaborate convergence acceleration is the fastest method. Stagnation of GMRES(3) in the B and RS cases is accompanied by small α_1 and oscillatory α_2 and α_3. mGMRES(3) avoids the stagnation of GMRES(3) and is superior to RK(3) in all cases.

The memory requirements of the 3D (m)GMRES(3) implementation on CRAY X-MP increased by 27% in comparison to the code for RK(3). This is essentially explained by the need to store three extra vectors.

The last example is another standard transonic test case: the ONERA M6 wing at $\alpha = 3.06°$ and $M_\infty = 0.84$. The grid with $120 \times 16 \times 32$ cells and the isobar pattern at the symmetry plane are depicted in Fig. 8. The parameter θ is 1.0 and the convergence results are found in Table V. The number of iterations required for convergence by RK(3) and mGMRES(3) are almost the same in the cases B, RS, and MG2 but mGMRES(3) is more demanding in CPU-time. GMRES(3) and RK(3) with MG1 did not converge. The α_1 parameter for GMRES(3) drops down to zero at the beginning of the stagnation phase in Fig. 9. The α_i parameters of mGMRES(3) vary smoothly with n in Fig. 10 (cf. Figs. 3 and 4). By letting $\varepsilon = 1.0\text{E}-6$ the number of iterations increased to 121 for RK(3) and MG2, but the calculated lift of the wing was the same in the first five digits. With $\theta = 0.5$ the number of iterations was 153 and the lift changed by 0.13%.

5. CONCLUSIONS

A number of numerical experiments with the stationary Euler equations have been made. The convergence rate for reaching the steady state solution is improved considerably by techniques that enhance the propagation of smooth error modes out from the computational domain. The stagnation problems with GMRES(3) disappear with mGMRES(3). The residual smoothing preconditioner in [5] is constructed to have an effect only on the smooth waves. When it is applied the total work for convergence is reduced. Addition of multigrid iteration also improves the convergence rate, in particular when more smoothing iterations are performed on the coarse grids. On these grids only the smooth error modes are represented and obtain an increased wave speed (and better damping).

mGMRES(3) is often more efficient than RK(3) in terms of number of iterations but not always if the CPU-seconds are compared. The extra cost per iteration for mGMRES(3) compared to RK(3) consists of one extra computation of $r(u)$ and more overhead. If that could be reduced by a more efficient implementation then mGMRES(3) would be the preferred method. However, for the 2D and 3D subsonic channel flow in Tables I and IV mGMRES(3) beats RK(3) also in CPU seconds. The reason is probably the adaptive damping properties of mGMRES. The experience from linear problems in [12] is similar. Also mGMRES(3) appears to be more robust than RK(3) in combination with multigrid iterations in our examples (see the discussion of the supersonic channel example and Table V). However, more numerical experiments are needed to assess the merits of mGMRES(k) compared to RK(k). Such work is in progress with $k > 3$ and for the Navier–Stokes equations.

ACKNOWLEDGMENTS

Bo-Erik Eriksson implemented GMRES and mGMRES and Mattias Steinwall coded the residual smoothing algorithm. Mattias Sillén made the grids around the airfoil and the wing.

REFERENCES

1. Å. Björck, Least squares methods, in *Handbook of Numerical Analysis, Vol. I*, edited by P. G. Ciarlet and J. L. Lions (North Holland, Amsterdam, 1990), p. 465.
2. A. Brandt, *Math. Comput.* **31**, 333 (1977).
3. M. O. Bristeau, R. Glowinski, B. Mantel, J. Périaux, and G. Rogé, in *Numerical Methods for Fluid Dynamics III*, edited by K. W. Morton and M. J. Baines (Clarendon Press, Oxford, 1988), p. 255.
4. P. Brown and Y. Saad, *SIAM J. Sci. Statist. Comput.* **11**, 450 (1990).
5. R. Enander, Ph.D. thesis, Dept. of Scientific Computing, Uppsala University, 1991.
6. R. Enander, *J. Comput. Phys.* **107**, 291 (1993).
7. B.-E. Eriksson, Report L-0-1 R143, SAAB-SCANIA, Linköping, Sweden, 1992 (unpublished).
8. L. Ferm, Report 139, Dept. of Scientific Computing, Uppsala University, 1991 (unpublished).
9. R. W. Freund, G. H. Golub, N. M. Nachtigal, *Acta Numer.* p. 57 (1992).
10. B. Gustafsson and P. Lötstedt, in *Proceedings, Fourth Copper Mountain Conf. on Multigrid Methods*, edited by J. Mandel *et al.*, (SIAM, Philadelphia, 1989), p. 181.
11. B. Gustafsson and P. Lötstedt, in *Proceedings, ICFD Conference on Numerical Methods for Fluid Dynamics, Reading, 1992.*
12. B. Gustafsson and P. Lötstedt, *Appl. Numer. Math.* **12**, 135 (1993).
13. W. Hackbusch, *Multigrid Methods and Applications* (Springer-Verlag, Berlin/Heidelberg, 1985).
14. A. Jameson, *Appl. Math. Comput.* **13**, 327 (1983).

15. A. Jameson, *Commun. Pure Appl. Math.* **41**, 507 (1988).
16. A. Jameson, W. Schmidt, and E. Turkel, AIAA Paper 81-1259, 1981 (unpublished).
17. R. Karlsson, Report LiTH-MAT-R-1990-11, Dept. of Mathematics, Linköping University, 1990 (unpublished).
18. P. Lötstedt, *SIAM J. Numer. Anal.* **29**, 1370 (1992).
19. P. Lötstedt and B. Gustafsson, *Math. Comput.* **60**, 473 (1993).
20. D. J. Mavriplis, *AIAA J.* **28**, 213 (1990).
21. N. M. Nachtigal, L. Reichel, and L. N. Trefethen, *SIAM J. Matrix Anal. Appl.* **13**, 796 (1992).
22. R.-H. Ni, *AIAA J.* **20**, 1565 (1982).
23. Y. Saad and M. H. Schultz, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
24. Y. C.-J. Sedin, U. G. Nävert, and A. Ålund, *Math. Comput. Simulation* **29**, 503 (1987).
25. M. Steinwall, Report L-0-1 R139, SAAB-SCANIA, Linköping, Sweden, 1991.
26. V. Venkatakrishnan, *AIAA J.* **29**, 1092 (1991).
27. H. F. Walker, *SIAM J. Sci. Stat. Comput.* **9**, 152 (1988).
28. L. B. Wigton, N. J. Yu, and D. P. Young, AIAA Paper 85-1494, 1985 (unpublished).