JOURNAL OF COMPUTATIONAL PHYSICS 83, 303–323 (1989)

A New Multigrid Approach to Convection Problems

WIM A. MULDER*

Department of Mathematics, 405 Hilgard Avenue,
University of California, Los Angeles, Los Angeles, California 90024-1555

Received April 1, 1988

A new multigrid method for convection problems is presented. It is designed to overcome the problem of alignment, the flow being aligned with the grid. The technique employs semi-coarsening in several directions simultaneously and gives rise to multiple coarser grids with the same total number of points per grid-level, but with different sizes in each co-ordinate direction. The amount of work per multigrid cycle is still $O(N)$. As an example, the method is applied to the nonlinear upwind-differenced Euler equations of gas dynamics in two dimensions. Convergence rates are estimated by two-level Fourier analysis for the linearised equations. Numerical experiments on the nonlinear equations confirm these estimates.

1. INTRODUCTION

The Euler equations of gas dynamics can be discretised in space by central or upwind differencing. The former requires artificial viscosity to avoid oscillatory solutions; the latter automatically introduces sufficient viscosity to avoid oscillations and allows for an excellent representation of shocks (see [18] and references therein). Although upwind schemes appear to be superior to central schemes in their representation of shocks and robustness, their cost is 4 to 8 times that of central schemes. For steady flows, this will not be a serious drawback if solutions can be obtained in only a few iterations.

Near the end of 1982, the author and van Leer investigated the efficiency of various relaxation schemes for the upwind differenced Euler equations [22]. A similar study was carried out independently by Chakravarthy [3]. In both papers, a significant acceleration with respect to conventional methods such as ADI and approximate factorisation was found. However, the convergence factors of these schemes are still only $1 - O(h^2)$.

The success of the relaxation methods inspired a multigrid approach. Convergence rates independent of grid size were found in early 1983 for a transonic test problem with a shock [9]. Both first-order- and second-order-accurate solutions

* This work has been supported in part by the Center for Large Scale Scientific Computing (CLaSSiC) Project at Stanford under ONR Grant N00014-82-K-0335, and in part by NSF Grant DMS85-03294 and ONR Grant N00014-86-K-0691 at UCLA.

0021-9991/89 $3.00

were computed. For the latter, the defect correction technique was used. The method was successfully applied to a demanding problem with strong shocks: the flow of interstellar gas in a spiral galaxy [10].

Jameson [7] introduced a multigrid method for a central-difference discretisation of the Euler equations around the same time. A method for the upwind differenced equations, in a number of respects similar to the one in [9], was proposed independently by Jespersen [8].

Several authors have applied and extended the method proposed by Jespersen [8] and Mulder [9]. A nonlinear version has been explored by Hemker and Spekreijse [6]. Three-dimensional computations have been carried out by Anderson [1]. Extensions to the Navier–Stokes equations can be found in [19, 20].

Despite these successes, the method sometimes fails. The problem is exposed in [13], where the convergence factors for various relaxation schemes and multigrid are estimated by two-level Fourier analysis. Convergence is lost in the case of *strong alignment*, the flow being aligned with the grid, which is a well-known problem already for elliptic equations with strong anisotropy [2, 4].

This paper describes a new method that does not suffer from poor convergence rates if alignment occurs. It employs semi-coarsening in d co-ordinate directions simultaneously, where d is the number of space dimensions. Here we will mainly concentrate on the 2-dimensional case. The generalisation to three dimensions is obvious.

Section 2 describes the method in a general way, without reference to a particular partial differential equation. Complexity estimates are included. The method is applied to the nonlinear Euler equations of gas dynamics in Sections 3 and 4. Convergence rates are estimated by two-level Fourier analysis on the linearised equations with constant coefficients and periodic boundary conditions in Section 3. Numerical experiments on the nonlinear equations, using upwind differencing, are carried out in Section 4 for a subsonic, transonic, and supersonic problem. The main conclusions are summarised in Section 5.

2. Method

2.1. *Motivation*

The multigrid technique is an efficient numerical method for solving elliptic partial differential equations. Its convergence rate is independent of the number of grid points. In combination with successive grid refinement, or nesting, it can provide solutions with an iteration error smaller than the discretisation error in one to three multigrid cycles [2, 4]. Non-elliptic problems can also be handled, if sufficient numerical ellipticity is present. This means that, on the scale of the grid spacing, there is a reasonable amount of coupling between neighboring points or cells. This coupling is then exploited to remove oscillatory components of the error by a suitable smoother. Here the error is defined as the difference between the correct numerical solution and the current guess.

The multigrid method fails if this coupling becomes too small in one of the co-ordinate directions. We exclude the possibility of total decoupling at some point, which means that the solution is undetermined at that point. In a multi-dimensional problem, the coupling can become locally 1-dimensional. This is a common situation in anisotropic problems, such as the elliptic problem $u_{xx} + \varepsilon u_{yy} = 0$ for $\varepsilon \to 0$, or the problem of computing a steady solution to the hyperbolic convection equation $u_t + au_x + bu_y = 0$ for $b \to 0$, $a \neq 0$. In these two examples, the differential operator becomes 1-dimensional, namely, $u_{xx} = 0$ for the elliptic example and $au_x = 0$ for the hyperbolic problem. Thus the solution will be independent of y, unless some structure in the y-direction is imposed by the boundary conditions. A proper discretisation will reflect this by being independent from neighboring values in the y-direction. Suppose now that the error has an arbitrary structure in the x-direction and is highly oscillatory in the y-direction. Restriction to a coarser grid will involve some sort of averaging in both co-ordinate directions. The oscillatory y-component will cause cancellation, and as a result, this error will not show up on the coarser grid and cannot be removed by the multigrid method. It also cannot be removed by smoothing on the finest grid, as would normally be done, because there is no coupling in the y-component. This component actually *must* remain unaffected if the discrete operator reflects the character of the differential equation. The only way to remove the error is by some relaxation scheme acting along the x-direction. If the x-component of the error is smooth, this will be an inefficient process. For a relaxation scheme that used only local data, the communication of boundary data to the interior will take $O(N^{1/2})$ iterations, where $N^{1/2}$ is the number of points in the x-direction. Thus, grid-independent convergence rates are lost.

The problem of decoupling in one of the co-ordinate directions is well known. It is called *strong alignment* by Brandt [2]. A commonly used remedy is based on brute force: choose a relaxation scheme that is exact or almost exact in the case of alignment. Candidates are *global* relaxation schemes such as those based on line relaxation or ILU. For purely hyperbolic problems, lexicographical Gauss–Seidel is a global scheme as well. It becomes a direct solver if the ordering follows the flow.

The initial success of the multigrid method for the Euler equations of gas dynamics proposed in [9], and its failure in certain cases [13], can be explained in the above terms. The Euler equations are hyperbolic for supersonic flow and elliptic for subsonic. In [9], the relaxation scheme is symmetric Gauss–Seidel, which is a poor smoother but a fairly good single-grid solver, although its asymptotic convergence rate is still $1 - O(h^2)$. The multigrid method accelerates the scheme considerably. However, symmetric Gauss–Seidel cannot handle alignment for subsonic flows, and therefore the multigrid method breaks down in that case. In practical applications such as transonic flow around an airfoil, using an O-grid, alignment only occurs in small regions and the flow is close to supersonic, so the method works fine. In subsonic channel flow, on the other hand, alignment is likely to occur and convergence will be slow.

In two space dimensions, the problem of alignment can be overcome by using line relaxation. Damped alternating-direction line–Jacobi relaxation provides a

uniformly good asymptotic convergence rate of 0.526 per cycle [12]. A disadvantage of this method is the global linearisation of the residual, required for the line relaxation. Also, line-relaxation is generally restricted to simple domains, whereas a multigrid scheme with a relaxation scheme that uses only local data, can in principle be used on domains of arbitrary shape.

Another way to solve the problem of alignment is by using semi-coarsening; that is, one combines points or cells in the direction of the strongest coupling [2, 4]. Any oscillatory component of the error in the direction of weak (or no) coupling will then be brought to the coarser grid. This approach has not been pursued to its full potential. The method presented here uses semi-coarsening in such a way that global solvers are no longer required (except on the coarsest grid).

Semi-coarsening can be implemented in several ways. The simplest approach involves coarsening in alternating directions. As an example, we consider a finite-volume discretisation on a finest grid of 8×8 points. The coarser grid can be 4×8 points, and the next coarser grid 4×4 points, et cetera. This method still fails in the case of alignment. If the flow follows the x-direction, then the restriction to 4×8 points will not cause any problems, but the subsequence restriction to the 4×4 grid will still cause oscillatory errors in the y-direction to cancel, whereas smoothing in the y-direction cannot be achieved in the absence of coupling. Thus, this approach is fruitless.

A second approach is based on semi-coarsening in two direction simultaneously. From one finest grid of 8×8 points, two coarser grids, one of 4×8 and one of 8×4 points, are obtained. On the next level, four grids are created, et cetera. In this way, the total number of grids is doubled after each restriction, and the total number of points in all grids on each level remains constant. The amount of work for a V-cycle is $O(N \log_2 N)$, as is the storage requirement. This method may be attractive for 2-dimensional problems, certainly if tailored for a computer with vector capabilities or parallel processors. However, in three dimensions, the number of points increases on progressively coarser grids. For example, starting from a $8 \times 8 \times 8$ grid, one would go to three grids of sizes $4 \times 8 \times 8$, $8 \times 4 \times 8$, and $8 \times 8 \times 4$, respectively, thus increasing the number of points by $\frac{3}{2}$. Proceeding to still coarser grids will result in excessive storage and operations requirements. The method is clearly useless in three dimensions.

2.2. *An $O(N)$ Method*

The complexity can be reduced to $O(N)$ by choosing the sequence of grids shown in Fig. 1. Starting from 8×8, semi-coarsening produces a 4×8 and a 8×4 grid, just as in the previous approach. On the next level, however, the 4×4 grid combines information from both finer grids. This method requires a modification of the standard multigrid algorithm to account for the redundancy when similar information from different grids is combined. Before discussing this, we will present complexity estimates for the method.

Let the finest grid have $N_1 = 2^{M_1}$ points in the x-direction and $N_2 = 2^{M_2}$ points

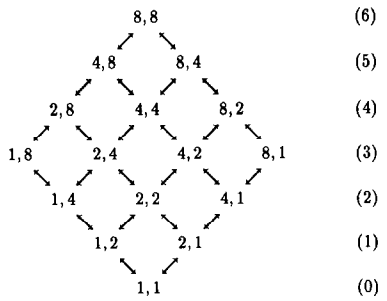

FIG. 1. Arrangement of finest (8×8) and coarser grids, that leads to an $O(N)$ multigrid method for problems with alignment. The level numbers are given on the right. The arrows indicate how the grids are linked by restriction (downward) and prolongation (upward).

in the y-direction, resulting in a total of $N = N_1 N_2$ points. The level number L corresponding to this grid is defined by $L = \log_2(N_1 N_2) = M_1 + M_2$. The set of grids corresponding to the finest grid consists of grids with size $2^{m_1} \times 2^{m_2}$, where $0 \leqslant m_1 \leqslant M_1$ and $0 \leqslant m_2 \leqslant M_2$. For each integer pair (m_1, m_2) there is one grid, and the total number of grids is obviously $(M_1 + 1)(M_2 + 1)$. Each grid (m_1, m_2) is linked to at most four grids $(m_1 \pm 1, m_2 \pm 1)$, two finer $(+1)$ and two coarser (-1). Less than four links occur if $m_1 \pm 1$ or $m_2 \pm 1$ lies outside the domain of m_1 and m_2, respectively.

The set of grids can be thought of as a lattice spanned by the integer pairs (m_1, m_2), with $0 \leqslant m_1 \leqslant M_1$ and $0 \leqslant m_2 \leqslant M_2$. Figure 1 illustrates the case $M_1 = M_2 = 3$. For a given level number l, with $0 \leqslant l \leqslant L$, the associated grids (m_1, m_2) are defined by $m_1 + m_2 = l$. The total number of points on all grids is

$$\sum_{m_1=0}^{M_1} \sum_{m_2=0}^{M_2} 2^{m_1} 2^{m_2} = 4N(1 - 2^{-M_1-1})(1 - 2^{-M_2-1}) \leqslant 4N. \tag{2.1}$$

For large M_1 and M_2, we need $4N$ points, i.e., four times the amount of storage required for the finest grid. The amount of work involved in a V-cycle is of the same order. Thus, we have an $O(N)$ cost per multigrid cycle, as in the usual multigrid approach.

A general multigrid cycle involves γ multigrid iteration with respect to a given grid. A V-cycle is obtained for $\gamma = 1$, a W-cycle for $\gamma = 2$. The cost of a cycle is approximately proportional to the number of points on each grid multiplied by the number of times that grid is visited. For a problem in d space dimensions, the finest grid has $N = N_1 N_2 \cdots N_d$ points. The cost of a multigrid cycle is roughly proportional to

$$\sum_{m_1=0}^{M_1} \cdots \sum_{m_d=0}^{M_d} 2^{m_1 + m_2 + \cdots + m_d} \gamma^{M_1 + \cdots + M_d - m_1 - \cdots - m_d}. \tag{2.2}$$

For a V-cycle ($\gamma = 1$), we obtain

$$2^d N \prod_{i=1}^{d} (1 - 2^{-M_i - 1}) \leqslant 2^d N,$$

which is the same as the total number of points on all grids. For a W-cycle ($\gamma = 2$), the result is

$$2^d N \prod_{i=1}^{d} (1 + M_i).$$

As $M_i = \log_2 N_i$, the $O(N)$ complexity is lost.

If a V-cycle does not solve the coarse-grid problem sufficiently well, an F-cycle can be used. This type of cycle is defined recursively as a $\gamma = 2$ multigrid cycle, in which the first multigrid iteration is an F-cycle and the second a V-cycle. The cost of an F-cycle can be estimated by

$$\begin{aligned} &\sum_{m_1=0}^{M_1} \cdots \sum_{m_d=0}^{M_d} 2^{m_1 + \cdots + m_d}(1 + M_1 + \cdots + M_d - m_1 - \cdots - m_d) \\ &\quad = 2^d N \left[(1+d) - \sum_{i=1}^{d} \frac{M_i + 1}{2^{M_i+1} - 1} \right] \left[\prod_{i=1}^{d} (1 - 2^{-M_i - 1}) \right] \\ &\quad \leqslant (1+d)\, 2^d N. \end{aligned} \tag{2.3}$$

We conclude that the proposed multigrid method has an $O(N)$ cost per multigrid cycle if a V- or F-cycle is used.

In this example we have used a finite-volume discretisation on a $2^{m_1} \times 2^{m_2}$ grids. For finite-difference discretisations, the number of points in each co-ordinate direction has to be increased by 1, i.e., we would have grids of size $(2^{m_1} + 1) \times (2^{m_2} + 1)$. The above results carry over in a straightforward manner to this situation.

2.3. *Restriction and Prolongation*

The restriction and prolongation operator for the method described above can be chosen in the same way as for a normal multigrid method that employs semi-coarsening. Only if data from more grids are to be transferred to one grid, does a modification have to be made. This modification is the crucial part of the present method. The description here is based on the full approximation storage (FAS) scheme [2] for the nonlinear system of equations $\mathcal{L}(u) = f$. The reader is assumed to be familiar with the details of the FAS scheme. The algorithm for linear equations follows from the nonlinear case in a straightforward way and will not be discussed here.

We start with the restriction operators. As mentioned in Section 2.2, a grid is uniquely defined by (m_1, m_2). The current solution on that grid is denoted by $u^{(m_1,m_2)}$, and the corresponding residual by $r^{(m_1,m_2)} = f^{(m_1,m_2)} - \mathcal{L}^{(m_1,m_2)}(u^{(m_1,m_2)})$. Let the restriction operator be denoted by R for the residual and by $\tilde{R}$ for the solution.

The initial solution on a coarser grid (m_1, m_2) on level $l = m_1 + m_2$ is found as the restriction of one or two solutions on level $l+1$, depending on the number of links to finer grids (cf. Fig. 1). Thus, the restriction operators R and $\tilde{R}$ correspond to semi-coarsening in the x-direction (R_x and $\tilde{R}_x$), or semi-coarsening in the y-direction (R_y and $\tilde{R}_y$), or a combination of both. The last is necessary if the coarser grid is linked to two finer grids. In that case, we give equal weights to the appropriately restricted data from each grid. An algorithmic description of the operations involved in restriction from level $l+1$ to l is:

for all (m_1, m_2) with $m_1 + m_2 = l$, $0 \leqslant m_1 < M_1$, $0 \leqslant m_2 < M_2$ do
  if there are links to $(m_1 + 1, m_2)$ and $(m_1, m_2 + 1)$ then

$$u^{(m_1,m_2)} := \tfrac{1}{2}[\tilde{R}_x u^{(m_1+1,m_2)} + \tilde{R}_y u^{(m_1,m_2+1)}]$$
$$\tau^{(m_1,m_2)} := \tfrac{1}{2}[R_x r^{(m_1+1,m_2)} + R_y r^{(m_1,m_2+1)}] - r^{(m_1,m_2)}(u^{(m_1,m_2)})$$

  else if there is only one link to $(m_1 + 1, m_2)$ then

$$u^{(m_1,m_2)} := \tilde{R}_x u^{(m_1+1,m_2)}$$
$$\tau^{(m_1,m_2)} := R_x r^{(m_1+1,m_2)} - r^{(m_1,m_2)}(u^{(m_1,m_2)})$$

  else if there is only one link to $(m_1, m_2 + 1)$ then

$$u^{(m_1,m_2)} := \tilde{R}_y u^{(m_1,m_2+1)}$$
$$\tau^{(m_1,m_2)} := R_y r^{(m_1,m_2+1)} - r^{(m_1,m_2)}(u^{(m_1,m_2)})$$

  end if
end do (2.4)

The resulting coarse-grid problem becomes

$$\mathscr{L}^{(m_1,m_2)}(\tilde{u}^{(m_1,m_2)}) = f^{(m_1,m_2)} + \tau^{(m_1,m_2)}.$$

Here the solution $\tilde{u}^{(m_1,m_2)}$ can be determined approximately by a multigrid cycle with respect to the grid (m_1, m_2), or by a single-grid solver if this grid is chosen to be the coarsest.

The basic difference with the standard multigrid approach is the occurrence of two links to finer grids. We have chosen equal weighting of data from the two grids. If $u^{(M_1-1,M_2)}$ and $u^{(M_1,M_2-1)}$ have been obtained directly by restriction from the finest grid (M_1, M_2), then

$$u^{(M_1-1,M_2)} = \tilde{R}_x u^{(M_1,M_2)}, \qquad u^{(M_1,M_2-1)} = \tilde{R}_y u^{(M_1,M_2)},$$
$$u^{(M_1-1,M_2-1)} = \tfrac{1}{2}[\tilde{R}_x \tilde{R}_y + \tilde{R}_y \tilde{R}_x]\, u^{(M_1,M_2)}. \tag{2.5}$$

A similar expression is obtained for the restriction of the residual. For common choices of the restriction operator, such as nine-point restriction or volume-weighted restriction, we have $\tilde{R}_{xy} = \tilde{R}_x \tilde{R}_y = \tilde{R}_y \tilde{R}_x$, implying that (2.5) results in the same expression as when $\tilde{R}_{xy}$ is applied directly. The same is true for all coarser

grids, that is, (2.4) produces the same result when going from the finest grid (M_1, M_2) to any arbitrary coarser grid (m_1, m_2), as one obtains by selecting an arbitrary path between these two grids and applying only $\tilde{R}_x$ or $\tilde{R}_y$. In practice, the solution will be modified on each grid before it is restricted to still coarser grids, and this equivalence disappears.

The prolongation operator brings corrections from one or more coarser grids to the current grid. We choose the following approach:

for all (m_1, m_2) with $m_1 + m_2 = l$, $0 < m_1 \leqslant M_1$, $0 < m_2 \leqslant M_2$ do

$$u^{(m_1,m_2)} := \mathcal{S}^{v_p}(u^{(m_1,m_2)}, f^{(m_1,m_2)}, \mathcal{L}^{(m_1,m_2)}(\cdot))$$

if there is a link to $(m_1 - 1, m_2)$ then

$$u^{(m_1,m_2)} := u^{(m_1,m_2)} + P_x[u^{(m_1-1,m_2)} - \tilde{R}_x u^{(m_1,m_2)}]$$

end if

if there is a link to $(m_1, m_2 - 1)$ then

$$u^{(m_1,m_2)} := u^{(m_1,m_2)} + P_y[u^{(m_1,m_2-1)} - \tilde{R}_y u^{(m_1,m_2)}]$$

end if

end do (2.6)

Here we have included v_p parallel smoothing steps with some relaxation scheme $\mathcal{S}$. Next the one or more coarse-grid corrections are added to the solution on the current grid (m_1, m_2). The fundamental difference with standard multigrid is that the corrections are always computed with respect to the restriction of the most recent solution rather than the one at the begin of the multigrid cycle. Choosing the x-direction first in (2.6) results in an asymmetry of the algorithm with respect to the co-ordinate directions. In the actual code described in Section 4, this asymmetry is reduced by alternating the order of prolongation in x and y each time (2.6) is carried out.

The restriction operator (2.4) uses equal weights when more than one finer grid exists. The prolongation operator (2.6) can also be expressed in terms of weights:

$$\begin{aligned} u^{(m_1,m_2)} := {} & [P_y u^{(m_1,m_2-1)}] + (I - P_y \tilde{R}_y)[P_x u^{(m_1-1,m_2)}] \\ & + (I - P_y \tilde{R}_y)(I - P_x \tilde{R}_x) \\ & \times [\mathcal{S}^{v_p}(u^{(m_1,m_2)}, f^{(m_1,m_2)}, \mathcal{L}^{(m_1,m_2)}(\cdot))] \end{aligned} \tag{2.7}$$

This shows that the weight for the first term on the right-hand side is I, for the second $(I - P_y \tilde{R}_y)$, and for the third $(I - P_y \tilde{R}_y)(I - P_x \tilde{R}_x)$. Because $P_x \tilde{R}_x$ and $P_y \tilde{R}_y$ approach I for the low frequencies, the low-frequency component of the new solution is mainly determined by $P_y u^{(m_1,m_2-1)}$. This preference with respect to one co-ordinate direction can be reduced by the alternating approach mentioned above. Thus, the next prolongation would result in an expression similar to (2.7), but with x and y interchanged.

In addition to the above parallel smoothing, one can perform ν_1 pre- and ν_2 post-smoothing steps, just as in the standard multigrid method. This completes the description of the new method.

Before we continue with the discussion of a specific application of the method, we like to point out that the choice of weights in (2.4) and (2.7) is convenient and simple, but not necessarily optimal. If these restrictions and (alternating) prolongation operators are used, no bias with respect to co-ordinate directions is introduced. Given the nature of the differential operator, which was assumed to be strongly biased with respect to one of the co-ordinate directions, it appears that a similar bias in the restriction and prolongation operators could give better results than the approach chosen here. These weights might possibly be based on norms of the residual on various grids, and on the associated convergence rates. The weights in (2.7) involve restriction and prolongation operators. One could include the differential operator if it is sufficiently simple. In addition, restriction to certain grids may skipped if there is only very weak coupling in the direction of coarsening. This is a viable option if the flow has a dominant direction throughout the computational domain and should save a substantial amount of work. Such variants remain to be explored.

Finally, we remark that the present method is not able to handle alignment at 45° with respect to the grid-lines. The discretisation used in the following sections does not recognise this as a special case and introduces sufficient numerical ellipticity to avoid the problem. Hackbusch [5] describes a parallel multigrid method which can handle alignment at 45° for certain simple problems. The price paid is a (sequential) complexity of $O(N \log N)$.

3. Two-Level Fourier Analysis for the Euler Equations of Gas Dynamics

The method sketched above is applied to the Euler equations of gas dynamics that describe the flow of an ideal compressible gas in the limit for vanishing viscosity. In conservation form, these are:

$$\frac{\partial w}{\partial t}+\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}=0. \tag{3.1a}$$

The vector of states w and the fluxes f and g are

$$w=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix}, \qquad f=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{pmatrix}, \qquad g=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vH\end{pmatrix}. \tag{3.1b}$$

Here ρ is the density of the gas, and u and v are the x- and y-components of the

velocity, respectively. The energy E, total enthalpy H, pressure p, and sound speed c are related by

$$E=\frac{1}{(\gamma-1)}\frac{p}{\rho}+\frac{1}{2}(u^2+v^2), \qquad H=E+\frac{p}{\rho}, \qquad c^2=\gamma\frac{p}{\rho}. \tag{3.2}$$

This system of four equations is discretised by upwind differencing, yielding solutions of first-order accuracy.

The convergence of the multigrid method can be estimated by linearising the equations (3.1), assuming the coefficients of the linearisation to be constant, and imposing periodic boundary conditions. This allows the use of Fourier symbols for the various operators. Then, only two levels are considered, a fine and a coarse, and it is assumed that the problem on the coarse grid is solved exactly. This procedure is followed in [13] for the standard multigrid method. For the present method, we choose a finest grid of size $N_1 \times N_2$, where $N_1 = 2^{M_1}$ and $N_2 = 2^{M_2}$ with both M_1 and M_2 larger than 0. It is assumed that the two coarser grids of size $(N_1/2)\times N_2$ and $N_1 \times (N_2/2)$, respectively, are solved exactly.

Here we only sketch the various steps in the two-level analysis. The present case is a straightforward generalisation of the work in [13]. The reader is referred to that paper for details.

A linearised form of (3.1) is given by the discrete operator

$$L=\frac{1}{h_x}[A^+(1-T_x^{-1})+A^-(T_x-1)] + \frac{1}{h_y}[B^+(1-T_y^{-1})+B^-(T_y-1)]. \tag{3.3}$$

Here h_x and h_y denote constant grid-spacings in the x- and y-directions, respectively. The shift operator T_x is defined by $T_x v_{k_1,k_2} = v_{k_1+1,k_2}$ for a quantity v on the $N_1 \times N_2$ grid, and $T_y v_{k_1,k_2} = v_{k_1,k_2+1}$. The matrices $A^\pm$ and $B^\pm$ correspond to the positive and negative parts of

$$A=\begin{pmatrix} u & 0 & c & 0 \\ 0 & u & 0 & 0 \\ c & 0 & u & 0 \\ 0 & 0 & 0 & u \end{pmatrix}, \qquad B=\begin{pmatrix} v & 0 & 0 & 0 \\ 0 & v & c & 0 \\ 0 & c & v & 0 \\ 0 & 0 & 0 & v \end{pmatrix}. \tag{3.4}$$

These matrices are obtained from df/dw and dg/dw after a similarity transform. The construction of these positive and negative parts is described in [13]. In Fourier space, the symbol $\hat{L}$ is a function of the frequencies

$$\theta_x = 2\pi l_1/N_1 \qquad (l_1 = -\tfrac{1}{2}N_1 + 1, \ldots, \tfrac{1}{2}N_1),$$
$$\theta_y = 2\pi l_2/N_2 \qquad (l_2 = -\tfrac{1}{2}N_2 + 1, \ldots, \tfrac{1}{2}N_2). \tag{3.5}$$

The singularities of $\hat{L}$ are listed in [13].

The restriction operator is based on volume-weighted averaging of the residuals and the solution, implying that $\tilde{R}_x = R_x = \frac{1}{2}(1 + T_x)$ and $\tilde{R}_y = R_y = \frac{1}{2}(1 + T_y)$. Prolongation is piecewise constant interpolation. In Fourier space the prolongation operators are the complex transposes of the restriction operators. These operators are the same as in [9, 13], but are now applied to two cells at the time, rather than four. The operator $\hat{R}_x$ introduces a coupling between θ_x and $\theta_x + \pi$, and $\hat{R}_y$ between θ_y and $\theta_y + \pi$. It is therefore convenient to consider the four frequency pairs (θ_x, θ_y), $(\theta_x + \pi, \theta_y)$, $(\theta_x, \theta_y + \pi)$, and $(\theta_x + \pi, \theta_y + \pi)$ simultaneously. With four equations, this requires 16×16 complex matrices in Fourier space. We omit the expressions for the symbols of the restriction and prolongation operators here, and merely outline the construction of the two-grid operators.

Let the linearised problem be given by $L^{(M_1, M_2)} u^{(M_1, M_2)} = f^{(M_1, M_2)}$, the right-hand side not to be confused with the flux in (3.1b). In terms of the error $v^{(M_1, M_2)} = \bar{u}^{(M_1, M_2)} - u^{(M_1, M_2)}$, which is defined as the difference between the converged solution $\bar{u}^{(M_1, M_2)}$ and the present guess, the relaxation operator is given by $v^{(M_1, M_2)} := S^{(M_1, M_2)} v^{(M_1, M_2)}$. The coarse-grid correction operators for the x- and y-directions are

$$\begin{aligned} K_x &= I - P_x (L^{(M_1 - 1, M_2)})^{\dagger} R_x, \\ K_y &= I - P_y (L^{(M_1, M_2 - 1)})^{\dagger} R_y. \end{aligned} \tag{3.3}$$

Here we have dropped the superscript (M_1, M_2). The dagger indicates the pseudo-inverse. The two-grid operator corresponding to (2.4) and (2.6) is given by

$$M_y = S^{\nu_2} [K_y + (I - P_y \tilde{R}_y) K_x + (I - P_y \tilde{R}_y)(I - P_x \tilde{R}_x)(S^{\nu_p} - I)] S^{\nu_1}. \tag{3.4}$$

The expression for M_x is obtained by interchanging x and y in (3.4). The corresponding two-grid convergence factors are

$$\lambda_x = \rho(\hat{L} \hat{M}_x \hat{L}^{\dagger}), \qquad \lambda_y = \rho(\hat{L} \hat{M}_y \hat{L}^{\dagger}). \tag{3.5}$$

Here $\rho(\cdot)$ denotes the spectral radius, which is computed from the Fourier symbols

included to account for its singularities. See [13] for details. For the Euler equations λ_x and λ_y are functions of u/c, v/c, N_1, N_2, h_y/h_x, θ_x, and θ_y. Note that u, v, c, h_x, and h_y are assumed to be constant. The maximum over the discrete set of frequencies is denoted by

$$\bar{\lambda}_x(u/c, v/c, h_y/h_x, N_1, N_2) = \max_{l_1, l_2} \lambda_x, \tag{3.7}$$

and a similar expression is obtained for $\bar{\lambda}_y$. The asymptotic convergence rate for two cycles is

$$\max \rho(\hat{L} \hat{M}_y \hat{M}_x \hat{L}^{\dagger}) \leqslant \max(\bar{\lambda}_x) \max(\bar{\lambda}_y) = (\max \bar{\lambda}_x)^2. \tag{3.8}$$

Here the maximum is taken over all admissible values of the five parameters. Because of symmetry, $\max \bar{\lambda}_x = \max \bar{\lambda}_y$.

The computation of $\bar{\lambda}_x(u/c, v/c, h_y/h_x, N_1, N_2)$ has has been carried out numerically. Figure 2 is the result of substantial number-crunching. It shows $\bar{\lambda}_x$ as a functions of u and v ($c=1$), for $h_y=h_x$. Each point represents the maximum over the finite set of frequencies that occur for a 64×64 grid. The smoother is damped point–Jacobi, which is given by

$$S^{PJ}=I-\frac{1}{2}N^{-1}L, \qquad N=\frac{1}{h_x}(A^+-A^-)+\frac{1}{h_y}(B^+-B^-). \tag{3.9}$$

This combination of the relaxation matrix N and damping by a factor $\frac{1}{2}$ completely removes the highest frequencies of $\hat{L}$ (at $\hat{T}_x=\hat{T}_y=-1$). For a 1-dimensional problem, this amount of damping yields an exact solver in combination with multigrid [11], and therefore seems to be a good choice for the present method. It should be noted that the relaxation matrix $N\geqslant 0$. Zero eigenvalues only occur for $u=v=0$. In that case, we use the pseudo-inverse in the linear analysis.

The worst convergence rate observed in Fig. 2 is $\frac{1}{2}$, and we conjecture that this is also the worst value of $\bar{\lambda}_x$ for all choices of u/c, v/c, h_y/h_x, $N_1\geqslant 2$, and $N_2\geqslant 2$. This statement is based on extensive computations.

Table I shows some results for three smoothers, the damped point–Jacobi mentioned above, damped red–black, where the damping is carried out in the same way as in (3.9), and damped symmetric Gauss–Seidel, with two sweeps in opposite directions. For the last, a nonstandard way of damping is employed [13, (6.13)]. Although Table I does not give the maximum over all parameters, it suggests that damped point–Jacobi is the best choice if only one pre- or post-smoothing step is performed. If only parallel smoothing is performed, one might consider red–black

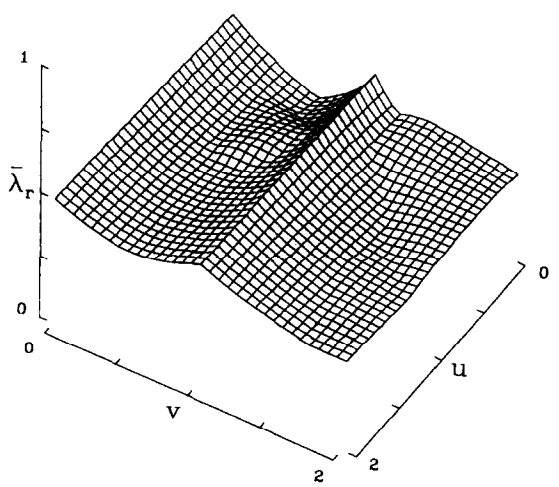

FIG. 2. Two-level convergence rate $\bar{\lambda}_x$ as a function of u and v ($c=1$), for a 64×64 grid with $h_x=h_y$, using damped Point–Jacobi smoothing with $\nu_1+\nu_2=1$, $\nu_p=0$. The worst convergence factor is $\frac{1}{2}$, for $v=0$ and $|v|=c$.

TABLE I

Two-Level Convergence Factors $\max(\bar{\lambda}_x)$ for the Damped Versions of Point–Jacobi, Red–Black, and Symmetric Gauss–Seidel Relaxation

Scheme	$\nu_1 + \nu_2$	ν_p	$\max(\bar{\lambda}_x)$
PJ	1	0	0.500
	2	0	0.500
	0	1	0.707
	0	2	0.707
RB	1	0	0.498
	2	0	0.535
	0	1	0.638
	0	2	0.587
SGS	1	0	0.499
	2	0	0.469
	0	1	0.574
	0	2	0.563

Note. The results are obtained for fixed $N_1 = N_2 = 64$ and $h_y = h_x$, and the maximum is taken over the values $u/c = -2 + n_1/8$, $v/c = -2 + n_2/8$, where both n_1 and n_2 are integers between 0 and 32.

or symmetric Gauss–Seidel. It should be noted that red–black smoothing is more costly than point–Jacobi, and that SGS is even more expensive in terms of cpu time. Increasing the number of smoothing steps from one to two improves the overall convergence factor, but, as seen can be seen from Table I, the worst convergence rate is hardly affected.

4. Numerical Experiments on the Nonlinear Euler Equations

4.1. *Implementation in the Nonlinear Case*

The linear two-level analysis provides an estimate of the convergence rate under ideal circumstances. If the method is applied to the nonlinear equations, convergence may be slower due to boundaries, large variations of solution values with respect to grid spacing, and nonlinear phenomena such as sonic lines, slip lines, and shocks, which may result in local singularities (cf. [11]). Another type of singularity is created by nonsmooth boundaries or grids. Ideally, the steady state Euler equations represent a pure boundary-value problem, with a solution that is identical to the one obtained from a time-accurate integration of the unsteady equations from arbitrary initial data, as time goes to infinity. Such an ideal situation is the exception rather than the rule, a fact one should be aware of when applying the method presented here, or any other implicit or Newton-like integration that is not time-

accurate and violates conservation in time. A steady state may not exist at all in some cases, or, what is more likely, multiple stationary solutions exist. A simple example is a shock in a straight channel, with supersonic inflow and subsonic outflow as boundary conditions: the position of the shock is undetermined, if this problem is considered as a pure boundary-value problem. Another example is the occurrence of regions in the flow where the velocity vanishes: for zero velocity and constant pressure, any density distribution will be steady. Stationary vortices are another problem. If multiple solutions exist, a steady-state solver may pick one that is unphysical, in the sense that it cannot be reached from any initial data, or that is unstable in time against small perturbations.

If the boundary-value problem has multiple solutions and a certain steady-state solver finds a stationary solution, then the solution may depend on the solver; that is, certain solvers may be unable to reach particular solutions. The solution will also depend on the discretisation scheme. In the case of zero velocity and constant pressure, for instance, a large amount of isotropic numerical viscosity will cause the solution to have constant density, which is not required by the differential equation. In general, it appears that Euler solvers provide acceptable results only because of numerical viscosity, where acceptable means that the results resemble the solution of the Navier–Stokes equations for small viscosity and thermal conductivity.

Bearing the above in mind, it is only reasonable to require fast convergence if the steady state is well defined. If convergence down to machine zero is desired, then the discrete equations should be sufficiently smooth around the steady state, otherwise the iterative process may end up in a limit cycle. For practical purposes, convergence to machine zero is a waste of time, but on the other hand, a limit cycle may prevent convergence to a solution within the accuracy of the discrete scheme, and it is therefore recommended to use smooth discretisations. The initial guess of the steady state is also important for fast convergence. It is usually obtained by successive grid refinement [2, 4]. A solution is computed on a coarse grid and interpolated to a finer one to obtain a good initial guess. In this way one needs a fixed number of iterations per grid to obtain a steady state in $O(N)$ operations. Computing the first solution on the coarsest grid may be a problem in itself. One can use time-stepping if necessary. Sometimes, the solution may change drastically when one goes to a finer grid. In that case, some type of continuation technique can be used. The initial guess can be obtained from a solution on the same finest grid, computed for a slightly different shape of the boundary, or slightly different boundary conditions.

We now turn to the details of the present scheme. The system of four conservation laws (3.1) is discretised by upwind differencing, using either van Leer's flux-vector splitting (FVS) [21] or Osher's scheme in the natural ordering [6, 15,16]. Both are sufficiently smooth for our purpose. A disadvantage of van Leer's scheme is its failure to recognize slip-lines and contacts. These are smeared out by the scheme. The reason is that van Leer's FVS does not have the proper eigenvalues. This may also cause problems at boundaries.

The vector of state quantities w is represented by cell-averages $w_{i,j}$ on a grid

consisting of arbitrary quadrilaterals, the four corners of which are denoted by $(i-\frac{1}{2}, j-\frac{1}{2})$, $(i+\frac{1}{2}, j-\frac{1}{2})$, $(i+\frac{1}{2}, j+\frac{1}{2})$, and $(i-\frac{1}{2}, j+\frac{1}{2})$, and the corresponding sides by $(i, j-\frac{1}{2})$, $(i+\frac{1}{2}, j)$, $(i, j+\frac{1}{2})$, $(i-\frac{1}{2}, j)$. This configuration is sketched in Fig. 3. The discrete residual is

$$\begin{aligned} r_{i,j} = &-s_{i,j-1/2} T^{-1}_{i,j-1/2} f(T_{i,j-1/2} w_{i,j}, T_{i,j-1/2} w_{i,j-1}) \\ &- s_{i+1/2,j} T^{-1}_{i+1/2,j} f(T_{i+1/2,j} w_{i,j}, T_{i+1/2,j} w_{i+1,j}) \\ &- s_{i,j+1/2} T^{-1}_{i,j+1/2} f(T_{i,j+1/2} w_{i,j}, T_{i,j+1/2} w_{i,j+1}) \\ &- s_{i-1/2,j} T^{-1}_{i-1/2,j} f(T_{i-1/2,j} w_{i,j}, T_{i-1/2,j} w_{i-1,j}). \end{aligned} \tag{4.1}$$

Here $f(w_l, w_r)$ provides an approximate solution to the Riemann problem, through one of the two upwind schemes mentioned above. The first rotation matrix is given by

$$T_{i,j-1/2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi_{i,j-1/2} & \sin\phi_{i,j-1/2} & 0 \\ 0 & -\sin\phi_{i,j-1/2} & \cos\phi_{i,j-1/2} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{4.2a}$$

where

$$\begin{aligned} \cos\phi_{i,j-1/2} &= \frac{y_{i+1/2,j-1/2} - y_{i-1/2,j-1/2}}{s_{i,j-1/2}}, \\ \sin\phi_{i,j-1/2} &= -\frac{x_{i+1/2,j-1/2} - x_{i-1/2,j-1/2}}{s_{i,j-1/2}}, \\ s_{i,j-1/2} &= [(x_{i+1/2,j-1/2} - x_{i-1/2,j-1/2})^2 + (y_{i+1/2,j-1/2} - y_{i-1/2,j-1/2})^2]^{1/2}. \end{aligned} \tag{4.2b}$$

Here the outward normal is $(\cos\phi_{i,j-1/2}, \sin\phi_{i,j-1/2})^T$. The rotation matrices for the other sides follow in a similar way.

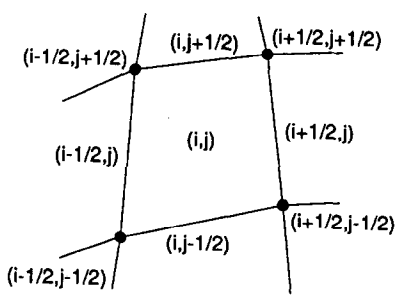

FIG. 3. Quadrilateral cell (i, j) of the computational domain showing the numbering of sides and vertices.

Given this discretisation, the next problem is to find a proper generalisation of the damped point–Jacobi smoother (3.9). The standard procedure for nonlinear equations is to apply Newton's method to solve $w_{i,j}$ from (4.1), keeping neighboring states fixed. Since only smoothing is required, one Newton iteration is usually sufficient [4]. The damping is applied afterwards.

With van Leer's flux vector-splitting, this is a feasible approach. In that case, the flux $f(w_l, w_r)$ is split as follows:

$$f(w_l, w_r) = f^+(w_l) + f^-(w_r), \qquad \frac{df^+(w_l)}{dw_l} \geqslant 0, \qquad \frac{df^-(w_r)}{dw_r} \leqslant 0. \tag{4.3}$$

The actual expressions for f^+ and f^- can be found in [21]. If Newton's method is used as part of the smoother, then we need $-\partial r_{i,j}/\partial w_{i,j}$. The first term on the right-hand side of (4.1) contributes

$$\frac{\partial}{\partial w_{i,j}} T^{-1}_{i,j-1/2} f(T_{i,j-1/2} w_{i,j}, T_{i,j-1/2} w_{i,j-1})$$

$$= T^{-1}_{i,j-1/2} \frac{df^+(\tilde{w})}{d\tilde{w}} T_{i,j-1/2}, \qquad \tilde{w} = T_{i,j-1/2} w_{i,j}. \tag{4.4}$$

Because positivity is preserved under a similarity transform, and the sum of positive matrices is positive, we conclude that for van Leer's flux-vector splitting,

$$N_{i,j} = -\frac{\partial r_{i,j}}{\partial w_{i,j}} \geqslant 0. \tag{4.5}$$

To avoid problems with zero eigenvalues, we add a small constant to the main diagonal, that is,

$$\tilde{N}_{i,j} = \frac{1}{\Delta t} - \frac{\partial r_{i,j}}{\partial w_{i,j}}, \tag{4.6}$$

where $1/\Delta t$ is chosen proportional to the norm of the residual (see Eq. (7.2b) in [13]). If only one Newton step is performed, the solution is updated according to

$$w_{i,j} := w_{i,j} + \tfrac{1}{2} \tilde{N}^{-1}_{i,j} r_{i,j}. \tag{4.7}$$

With Osher's scheme in the natural ordering, this positivity is lost. Only if w_l is close enough to w_r, we have $\partial f(w_l, w_r)/\partial w_l \geqslant 0$. In general, the matrix $N_{i,j} = -\partial r_{i,j}/\partial w_{i,j}$ may have eigenvalues that are negative, or even complex. This may lead to disastrous results. It is quite surprising that the authors of [6] obtained fairly good results with this approach. A particular source of trouble is a hard wall. In [6], the flux at a vertical wall at the right side of a cell is computed by $f_B = (0, p_B, 0, 0)^T$, where the pressure at the boundary is determined from characteristic variables:

$$c_B = c + \frac{\gamma - 1}{2} u, \qquad \rho_B = r(c_B/c)^{2/(\gamma-1)}, \qquad p_B = \rho_B c_B^2/\gamma. \tag{4.8}$$

The same result is obtained if a mirror zone with reflected states is introduced at the other side of the wall (same states, but the normal velocity gets a minus sign), and the Riemann problem at the wall is solved by Osher's scheme in the natural ordering—but only if the normal velocity is subsonic. If one takes the Jacobian of the wall flux, and if the wall is not perpendicular to the other sides of the cell, then the contribution to the relaxation matrix may be negative, even if no strong gradients are present. In the numerical experiments described below, our code started to produce nonsense if one cell had non-parallel walls on both sides and the flow was close to supersonic.

To avoid this problem, we choose an inconsistent linearisation. Instead of the proper Jacobian, we use

$$T^{-1}_{i,j-1/2} A^+(\tilde{w}_{i,j}) T_{i,j-1/2}, \qquad \tilde{w}_{i,j} = T_{i,j-1/2} w_{i,j} \tag{4.9}$$

for the first side of the quadrilateral and similar expressions for the other sides. Here A^+ is determined from $A = \partial f(w)/\partial w$ by selecting the positive eigenvalues:

$$A^+ = Q^{-1} \begin{pmatrix} \max(u-c,0) & 0 & 0 & 0 \\ 0 & \max(u,0) & 0 & 0 \\ 0 & 0 & \max(u,0) & 0 \\ 0 & 0 & 0 & \max(u+c,0) \end{pmatrix} Q, \tag{4.10a}$$

where

$$Q = \begin{pmatrix} 1 & 0 & 1 & 1 \\ u-c & 0 & u & u+c \\ v & 1 & v & v \\ H-uc & v & \frac{1}{2}(u^2+v^2) & H+uc \end{pmatrix}. \tag{4.10b}$$

In this way, $N_{i,j} \geqslant 0$, and the modification (4.6) avoids zero eigenvalues.

The boundary conditions at a hard wall are implemented by means of a mirror cell with reflected states and using the approximate Riemann solver in the same way as in the interior. In computing A^+, the dependence of the mirror states on the interior states is ignored. At open boundaries, we determine states on the exterior side of the boundary by selecting 1-dimensional characteristic variables either from free-stream values or from the interior, according to the sign of their corresponding eigenvalues. Then the approximate Riemann solver is used to determine the flux across the boundary, in the same way as in the interior. The dependence of the exterior values on the interior ones is ignored when the Jacobians are computed.

It is not clear a priori if damping should be used at the boundaries or not. In assembling the relaxation matrix N, one may include the damping right away; that is, one assembles $\frac{1}{2}N$ rather than N. At boundaries, one might suppress the damping by including the full contribution from the side of the quadrilateral that coincides with the boundary, rather than half of it. Analysis for the 1-dimensional scalar

equation $u_t + au_x = 0$ shows that for point–Jacobi it is better to include damping at open boundaries. Thus, an open boundary (inflow or outflow) is treated in the same way as the interior. The numerical experiments described in the next section indicate that it is slightly better to use the damping at hard walls as well. In brief, the construction of N at the boundaries is identical to its construction in the interior of the computational domain.

4.2. *Experiments*

As a test problem we consider flow through a channel with a circular arc at the bottom, with subsonic, transonic, and supersonic steady solutions. The grid is shown in Fig. 4a. It is practically equidistant in the y-direction and also in the x-direction across the bump. Constant stretching has been applied in the horizontal direction away from the bump. The finest grid use in the computation is actually 128×64, but that is too fine to be displayed without Moiré patterns. The dimensions of the grid are 5 by 2, with the circular arc between 1.5 and 2.5, having a thickness of 4.2% of the chord [17].

The multigrid solver uses a coarsest grid of 4×2 cells. The equations on the coarsest grid are solved by four sweeps of (undamped) symmetric Gauss–Seidel. The multigrid technique is implemented as a full approximation storage scheme [2], and the initial guess is obtained by successive grid-refinement. An F-cycle is used to solve the equations on each grid. For problems with shocks, it is recommended to have post-smoothing. The reason is that the coarse-grid corrections can be large around the shock, if the shock is in the wrong position. These large corrections introduce large high-frequency errors that have to be removed immediately by post-smoothing, otherwise the entire multigrid process will go astray. On the basis of Table I we choose one post-smoothing step with damped point–Jacobi.

On the finest grid, one smoothing step is sufficient in smooth regions of the flow, but not near shocks, where the coarse-grid correction may introduce $O(1)$ high-frequency errors. Therefore, one additional local relaxation step with (undamped) symmetric Gauss–Seidel at 4×4 cells around the largest residual is performed at the end of each cycle (cf. [11]). This requires hardly any extra work and improves the overall convergence rate substantially in some cases. Because an F-cycle is used, smoothing is applied more than once on the coarser grids. In the experiments done here, local relaxation does not appear to be necessary on the coarser grids, although in general it may be.

Local relaxation is also helpful after grid refinement. In that case we use third-order interpolation (see [11]). At shocks, and also near non-smooth parts of the boundary, large local errors may occur, that are easily removed by local relaxation. Again we use only one symmetric Gauss–Seidel sweep on 4×4 cells.

The results shown in Figs. 4b–d have been obtained with an F-cycle. The finest grid has 128×64 cells, the coarsest grid 4×2. Convergence rates for the cases shown in Fig. 4 vary between 0.3 and 0.4, well below the worst value predicted by the two-level analysis. These results are obtained both for van Leer's flux-vector splitting and Osher's scheme in the natural ordering. For the first, a consistent

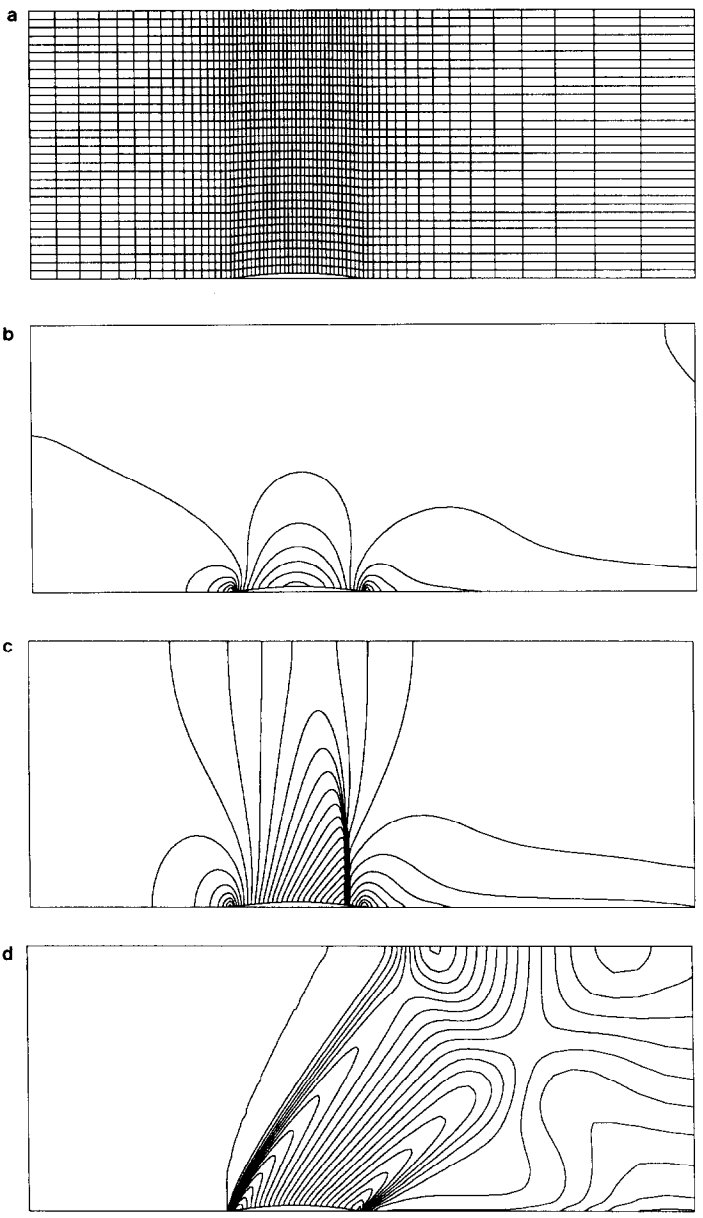

FIG. 4. (a) 64×32 grid used in the computations. The bump at the bottom at the channel has a thickness of 0.042. Inflow is from the left, outflow to the right. The results shown in the following figures are obtained on a 128×64 grid. (b) Iso-Mach lines for mach 0.5 inflow. Contours are 0.01 apart (first-order Osher scheme). The asymmetry gives an indication of the low accuracy of the first-order scheme. (c) Iso-Mach lines for mach 0.85 inflow. Contours are 0.025 apart (first-order Osher scheme). (d) Iso-Mach lines for mach 1.4 inflow. Contours are 0.025 apart (first-order Osher scheme).

linearisation (4.4) is used for the relaxation matrix $\tilde{N}$ (4.6), except near boundaries where the dependence of the exterior quantities on the interior states is ignored. For Osher's scheme, the inconsistent linearisation (4.9) is adopted.

An initial guess of the steady solution is obtained by the nested iteration technique [5], which is known as *full multigrid* when used in combination with a multigrid solver [2]. Here we first perform a fixed number of cycles on a 8×4 grid, then interpolate the solution by piecewise linear interpolation to a 16×8 grid, again perform a fixed number of multigrid iterations, and so on. Note that, for this nesting, the number of cells is doubled in both co-ordinate directions at the same time. According to [4, Eq. (5.2.9)], one or two cycles should be sufficient to obtain a solution with an iteration error of the same size as the discretisation error. In the presence of shocks and non-smooth boundaries, more iterations may be required to prevent large local iteration and discretisation errors from contaminating the rest of the solution. In the experiments, we stopped iterating if the L_∞ norm of the actual

tions. This generally took about six multigrid cycles and produces results with an iteration error well below the discretisation error.

It should be noted that the solutions obtained in this way have only first-order accuracy. Higher order accuracy is desirable for practical applications. This will be considered elsewhere [14].

5. Conclusions

A new multigrid method for problems with alignment has been presented. Its complexity is $O(N)$ if a V- or F-cycle is used. The method has a fair amount of parallelism, due to the use of several coarser grids on the same level of coarseness, and the optional use of smoothing parallel to the computation of the coarse-grid correction. This parallelism has not been explored in this paper.

Because the method combines data of more grids into one, the standard restriction and prolongation operators had to be modified. The choise of restriction and prolongation operator presented here is convenient, but it may be possible to obtain better convergence rates with other choices.

The application to the Euler equations of compressible gas dynamics shows that the method has uniformly good convergence rates, both in the linear two-level analysis and in the nonlinear subsonic, transonic, and supersonic experiments. With damped point–Jacobi used for post-smoothing, the two-level analysis predicts a worst convergence factor of $\frac{1}{2}$, whereas the numerical experiments provide values between 0.3 and 0.4.

Epilogue

The present work has been inspired by ideas on parallel multigrid. The numerical computations for the two-level analysis were carried out on the four processor Alliant at the Department of Computer Science at Stanford University, and on the 32 processor Ncube at UCLA. The nonlinear experiments

were performed on a SUN 3/110 work station at UCLA. Figure 1 has been typeset in T_EX, Fig. 2 is drawn by the NCAR-graphics package, Fig. 3 has been composed in PostScript, and Figs. 4a–d have been produced with the package NUTOIS, written by Tom Smedsaas from the Department of Scientific Computing at Uppsala University.

REFERENCES

1. W. K. ANDERSON, Ph.D. thesis, Mississippi State University, 1986 (unpublished).
2. A. BRANDT, in *Lecture Notes in Mathematics Vol. 960* (Springer-Verlag, Berlin/Heidelberg/New York, 1982), p. 220.
3. S. R. CHAKRAVARTHY, "Relaxation Methods for Unfactored Implicit Upwind Schemes," AIAA Paper 84–0165, AIAA 22nd Aerospace Sciences Meeting, Reno, Nevada, Jan. 1984.
4. W. HACKBUSCH, *Multi-Grid Methods and Applications*, Computational Mathematics Vol. 4 (Springer-Verlag, Berlin/Heidelberg/New York, 1985).
5. W. HACKBUSCH, in *Proceedings, First International Conference on Industrial and Applied Mathematics*, edited by J. McKenna and R. Temam (SIAM, Philadelphia, 1988).
6. P. W. HEMKER AND S. P. SPEKREIJSE, *Appl. Numer. Math.* **2**, 475 (1986).
7. A. JAMESON, *Appl. Math. Comput.* **13**, 327 (1983).
8. D. C. JESPERSON, *Appl. Math. Comput.* **13**, 357 (1983).
9. W. A. MULDER, *J. Comput. Phys.* **60**, 235 (1985).
10. W. A. MULDER, *Astron. Astrophys.* **156**, 354 (1986).
11. W. A. MULDER, "Multigrid for the One-Dimensional Inviscid Burgers' Equation," to appear in *SIAM J. Sci. Stat. Comput.*
12. W. A. MULDER, A note on the use of symmetric line Gauss–Seidel for the steady upwind differenced Euler equations, *SIAM J. Sci. Stat. Comput.*, in press.
13. W. A. MULDER, "Analysis of a Multigrid Method for the Euler Equations of Gas Dynamics in Two Dimensions," in *Multigrid Methods: Theory, Applications, and Supercomputing*, edited by S. McCormick (Dekker, New York, 1988).
14. W. A. MULDER, "A High-Resolution Euler Solver," AIAA Paper 89-1949, AIAA 9th Computational Fluid Dynamics Conference, Buffalo, NY, June 1989.
15. S. OSHER AND F. SOLOMON, *Math. Comput.* **38**, 339 (1982).
16. M. M. RAI AND S. R. CHAKRAVARTY, *AIAA J.* **24**, 735 (1986).
17. A. RIZZI AND H. VIVIAND (Eds.), in *Proceedings, GAMM Workshop, Stockholm, 1979* (Vieweg Verlag, Brunswick/Munich, 1981).
18. P. L. ROE, *Ann. Rev. Fluid Mech.* **18**, 337 (1986).
19. G. SHAW AND P. WESSELING, in *Lecture Notes in Physics, Vol. 264* (Springer-Verlag, Berlin/Heidelberg/New York, 1986), p. 566.
20. W. SCHRÖDER AND D. HÄNEL, in *Lecture Notes in Mathematics Vol. 1228* (Springer-Verlag, Berlin/Heidelberg/New York, 1986), p. 272.
21. B. VAN LEER, in *Lecture Notes in Physics Vol. 170* (Springer-Verlag, Berlin/Heidelberg/New York, 1982), p. 507.
22. B. VAN LEER AND W. A. MULDER, in *Numerical Methods for the Euler Equations of Fluid Dynamics*, edited by F. Angrand, A. Dervieux, J. A. Desideri, and R. Glowinski (SIAM, Philadelphia, 1985), p. 312.