

# Multigrid Approaches to the Euler Equations

P. W. HEMKER AND G. M. JOHNSON

**3.1. Introduction.** In this chapter we give a survey of the present state of the art for multigrid solution of the Euler equations for inviscid compressible flow. This is an example of a branch of multigrid research in which a thorough mathematical basis is still missing. What does exist to guide applications is an abundance of heuristic arguments and analogues from areas with better theoretical foundations, yet there is a scarcity of solid theory to account for the convergence speed and efficiency shown in practice. Such theory as there is lags well behind both practical development and the excellent results that have already been obtained with multigrid Euler solvers applied to large scale problems.

We chose to restrict our discussion in this chapter to the Euler equations because there are a few visible lines of development that can easily be treated within the scope of this chapter. Much interesting work has also been done in the general field of compressible and incompressible Navier–Stokes equations (cf. the pioneering work by Brandt [92], [95], [99], [105]), but the state of the art in this area is changing too rapidly to be suitable for discussion here. We refer the reader instead to the literature for other fluid flow applications. In particular, the KWIC index to the Multigrid Bibliography included in Appendix 2 of this book lists a collection of papers on the compressible and incompressible Navier equations, potential flow, and the Stokes equations.

Because even the multigrid Euler-solver discipline is continually changing, in this chapter we adopt the perspective of an overview rather than one of prescription and detailed guidance. We hope that this overview and the cited references will prepare the reader for further studies in this advancing field.

The efficient solution of flow problems was one of the early aims in the applications of multigrid (MG) methods [86]. However, in recent years most of the progress in the development of MG has been made in the field of elliptic partial differential equations and other fields where a solid mathematical theory exists (e.g., integral equations). For the inherently more

complex equations that describe flow problems, the theoretical development of MG did not proceed at the same pace. Early numerical work was done by Brandt [99], [105] and South and Brandt [524], where, for example, the Stokes equations and the incompressible and compressible Navier–Stokes equations were considered.

On the other hand, triggered by practical interest from the engineering sciences, several attempts have been made to apply MG ideas to improving the efficiency of flow computations. If the flow is assumed to be irrotational, then it can best be described by the potential equation, which—in the interesting case of transonic flow—is of mixed hyperbolic and elliptic type. By the use of MG, substantial improvements were made in the procedures for solving these equations [70], [123], [297], [382], [439], [524]. When the assumption of irrotational flow is dropped, an exact description of inviscid flow is given by the Euler equations. When the physical effects of viscosity and heat conduction are also included, these equations extend to the Navier–Stokes equations. Models of turbulence can also be included in the Navier–Stokes equations.

In this chapter we will treat several multiple grid approaches that are used for the solution of the equations of compressible flow. We restrict ourselves to problems in 2 space dimensions. Almost all techniques discussed here can be applied in 3-D as well, but the burden of 3-D notation makes the description unattractive. Also, in practice, most codes are written for 2-D problems because the complexity of 3-D computations and the computational requirements for their implementation are at the limit of present-day computer capabilities. The advent of more powerful computers will certainly change this situation in the near future.

Although practical problems that arise in the aircraft and turbomachinery industries are often described by the compressible Navier–Stokes equations, we shall consider mainly the Euler equations of inviscid flow. The reason for this is the assumption that a good method for the solution of the Euler equations may be extended to those situations where viscosity plays a significant role.

In those cases where the solution of the Euler equations can be used as a first approximation to the solution of the full Navier–Stokes equations, it may be a convenient approach to compute (an approximation to) this Euler flow first. This approximation can then be corrected for viscous effects. Most simply, this is done by a defect correction approach [66], where the solution of the Navier–Stokes equations is found by an iterative process in which only Euler-type equations are (approximately) solved and the heat conduction and viscous Navier–Stokes terms are taken care of by adding the corresponding corrections as forcing terms. In practice, a simple method to realize such an iterative process for the solution of the Navier–Stokes equation is to neglect the extra Navier–Stokes terms at particular stages of the solution process.

**3.1.1. The equations.** The 2-D *Navier–Stokes equations*, describing the physical laws of conservation of mass, momentum and energy, can be written in conservation form as

$$(1.1a) \quad \frac{\partial}{\partial t} \mathbf{q} + \frac{\partial}{\partial x} F(\mathbf{q}) + \frac{\partial}{\partial y} G(\mathbf{q}) = 0,$$

where

$$(1.1b) \quad F(q) = f(q) - \text{Re}^{-1} r(q), \quad G(q) = g(q) - \text{Re}^{-1} s(q),$$

and

$$q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, \quad f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, \quad g = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{pmatrix},$$

$$r = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \kappa \text{Pr}^{-1} (\gamma - 1)^{-1} (c^2)_x + u\tau_{xx} + v\tau_{xy} \end{pmatrix},$$

$$s = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \kappa \text{Pr}^{-1} (\gamma - 1)^{-1} (c^2)_y + u\tau_{xy} + v\tau_{yy} \end{pmatrix}.$$

Here  $\rho$ ,  $u$ ,  $v$ ,  $e$  and  $p$ , respectively, represent density, velocity in  $x$ - and  $y$ -direction, specific energy and pressure;  $H = e + p/\rho$  is the specific enthalpy. The pressure is obtained from the equation of state, which for a perfect gas reads

$$p = (\gamma - 1)\rho(e - \frac{1}{2}(u^2 + v^2));$$

$\gamma$  is the ratio of specific heats.  $q(t, x, y)$  describes the state of the gas as a function of time and space and  $f$  and  $g$  are the convective fluxes in the  $x$ - and  $y$ -direction, respectively.  $\text{Re}$  and  $\text{Pr}$  denote the Reynolds and Prandtl numbers; thermal conductivity is given by  $\kappa$ ;  $c = \sqrt{\gamma p/\rho}$  is the local speed of sound; and

$$\tau_{xx} = (\lambda + 2\mu)u_x + \lambda v_y, \quad \tau_{xy} = \mu(u_y + v_x), \quad \tau_{yy} = (\lambda + 2\mu)v_y + \lambda u_x,$$

where  $\lambda$  and  $\mu$  are viscosity coefficients. Stokes assumption of zero bulk viscosity may reduce the number of coefficients by one:  $3\lambda + 2\mu = 0$ .

We denote the open domain of definition of (1.1) by  $\Omega^*$ .

The *Euler equations* are obtained from (1.1a) by neglecting viscous and

heat conduction effects:

$$(1.1c) \quad F(q) = f(q), \quad G(q) = g(q).$$

The time-dependent Euler equations form a hyperbolic system: written in the quasi-linear form

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial f}{\partial q} \cdot \frac{\partial \mathbf{q}}{\partial x} + \frac{\partial g}{\partial q} \cdot \frac{\partial \mathbf{q}}{\partial y} = 0,$$

the matrix

$$(1.2) \quad k_1 A + k_2 B = k_1 \frac{\partial f}{\partial q} + k_2 \frac{\partial g}{\partial q}$$

has real eigenvalues for all directions  $(k_1, k_2)$ . These eigenvalues are  $(k_1 u + k_2 v) \pm c$  and  $(k_1 u + k_2 v)$  (a double eigenvalue). The sign of the eigenvalues determines the direction in which the information about the solution is carried along the line  $(k_1, k_2)$  as time develops (i.e., it determines the direction of flow of characteristic information). It locates the direction of the domain of dependence.

It is well known that, because of the nonlinearity, solutions of the Euler equations may develop discontinuities, even if the initial flow ( $t = t_0$ ) is smooth. To allow discontinuous solutions, (1.1) is rewritten in its integral form

$$(1.3) \quad \frac{\partial}{\partial t} \iint_{\Omega} \mathbf{q} \, dx \, dy + \int_{\partial\Omega} (f \cdot n_x + g \cdot n_y) \, ds = 0 \quad \text{for all } \Omega \subset \Omega^*;$$

$\partial\Omega$  is the boundary of  $\Omega$  and  $(n_x, n_y)$  is the outward normal vector at the wall  $\partial\Omega$ .

The form (1.3) of equation (1.1) shows clearly the character of the system of conservation laws: the increase of  $q$  in  $\Omega$  can be caused only by the inflow of  $q$  over  $\partial\Omega$ . In symbolic form we write (1.3) as

$$(1.4) \quad \mathbf{q}_t + N(\mathbf{q}) = 0.$$

The solution of the weak form (1.3) of (1.1a, c) is known to be nonunique, and a physically realistic solution (which is the limit of a flow with vanishing viscosity) is known to satisfy the additional entropy condition (cf. [La1], [La2]). The entropy condition implies that characteristics do not emerge at a discontinuity in the flow.

The *steady state* equations are obtained by the assumption  $\partial \mathbf{q} / \partial t = 0$ . Guided by the defect correction principle and knowing how the viscous effects change the governing equations, for the Navier–Stokes equations with large Reynolds number we can concentrate on the solution methods for

the stationary Euler equations

$$(1.5) \quad N(\mathbf{q}) = 0.$$

**3.1.2. The discretizations.** For the discretization of (1.1) or (1.3), two different approaches can be taken. First, the time and space discretizations can be made at once. This leads, for example, to discretization schemes of Lax–Wendroff type. An initial state of the fluid  $q_{(n)}^h$ , defined on a discrete grid, is advanced over one time step. Using a second order approximation in time yields

$$(1.6) \quad q_{(n+1)}^h = q_{(n)}^h + \Delta t (q^h)_t + \frac{1}{2} (\Delta t)^2 (q^h)_{tt}.$$

With the equation (1.1a, c), we arrive at

$$q_{ij}^{(n+1)} = q_{ij}^{(n)} - \Delta t (f_x + g_y)_{ij} + \frac{1}{2} (\Delta t)^2 \left\{ [A(f_x + g_y)]_x + [B(f_x + g_y)]_y \right\}_{ij},$$

where  $A$  and  $B$  are defined by (1.2). Using various difference approximations of the bracketed terms in the right-hand side, we may obtain different Lax–Wendroff type discretizations.

This type of discretization is usually made on a rectangular grid. If the domain  $\Omega^*$  is not rectangular, a 1-1-mapping  $(x, y) \leftrightarrow (\xi, \eta)$  between the physical domain and a rectangular computational domain can be constructed. Then the differential equation and the boundary conditions are reformulated on this computational domain.

A property of most of these Lax–Wendroff discretizations is that, when by time stepping a stationary state is obtained such that  $q_{(n+1)}^h = q_{(n)}^h$ , the discrete stationary state still depends on  $\Delta t$ . This is caused by the fact that the discrete term with  $(\Delta t)^2$  in (1.6) in general does not vanish.

A second approach is to distinguish clearly between the time and the space discretization by the method of lines. First, a space discretization is made for the partial differential equation (1.4) by which it is reduced to the large system of ordinary differential equations

$$(1.7) \quad \frac{\partial}{\partial t} \mathbf{q}^h = N^h(\mathbf{q}^h).$$

Now, to find an approximation of the time-dependent solution of (1.4), any method can be used for the integration of this system of ordinary differential equations. The solution of the steady state can be computed by solving (1.7) until the transients have died out. Alternatively, we can avoid the ordinary differential equations (1.7) and solve the nonlinear system

$$(1.8) \quad N^h(\mathbf{q}^h) = 0$$

by other (more direct) means. In both cases (1.7) and (1.8), we find a steady approximate solution  $\mathbf{q}^h$  independent of the choice of a time step.

For the construction of the semi-discrete system (1.7) or (1.8) on a nonrectangular domain  $\Omega^*$ , a mapping  $(x, y) \leftarrow \rightarrow (\xi, \eta)$  can again be introduced and finite difference approximations (of arbitrarily high order) can be used to construct a space discretization of the transformed steady equation

$$[y_\eta F(\mathbf{q}) - x_\eta G(\mathbf{q})]_\xi + [-y_\xi F(\mathbf{q}) + x_\xi G(\mathbf{q})]_\eta = 0.$$

Another way to construct system (1.7) on a nonrectangular grid is by a *finite volume* technique. Here, the starting point for the discretization is (1.3). Without an a priori transformation, the domain  $\Omega^*$  is divided into a set of disjoint quadrilateral cells  $\Omega_{ij}$ . The discrete representation  $\mathbf{q}^h$  of  $\mathbf{q}$  is given by the values  $\mathbf{q}_{ij}$ , the (mean) values of  $\mathbf{q}$  in the cell  $\Omega_{ij}$ . Using different approximations for the computation of fluxes between the cells  $\Omega_{ij}$ , we obtain various finite volume discretizations. We can easily obtain a conservative scheme by computing a unique approximation for each flux over the boundary between two neighboring cells.

In order to define a proper sequence of discretizations as  $h \rightarrow 0$  for a nonrectangular grid, a formal relation between the vertices of cells  $\Omega_{ij}$  and a regular grid can be given, again by a mapping  $(x, y) \leftarrow \rightarrow (\xi, \eta)$ . If this mapping is smooth enough, it can be proved that, for refinements  $h \rightarrow 0$  corresponding to regular refinements in  $(\xi, \eta)$ , space discretizations up to second order can be obtained. An advantage of the finite volume technique is that the untransformed equations can be used, even for a complex region. Boundary condition information is also usually simpler for finite volume methods.

With the finite volume technique, both central difference and upwind type finite volume schemes are used. They differ by the computation of the flux between neighboring cells  $\Omega_{ij}$ :

(1) For a central difference type, the flux over a cell wall  $\Gamma_{LR}$  between two cells with states  $q_L$  and  $q_R$  is computed as  $\frac{1}{2}f^*(q_L) + \frac{1}{2}f^*(q_R)$ , where  $f^* = k_1 f + k_2 g$  is the flux normal to  $\Gamma_{LR}$ . On a Cartesian grid this scheme reduces to the usual central difference scheme. In order to stabilize this scheme, and to prevent the uncoupling of odd and even cells in the grid, it is necessary to supplement the scheme with some kind of artificial dissipation (artificial viscosity).

(2) For upwind difference type discretizations, numerical flux functions  $f^*(q_L, q_R)$  are introduced to compute the flux over  $\Gamma_{LR}$ . Several functions  $f^*$  are possible. They solve approximately the Riemann problem of gas dynamics: they approximate the flux between two (initially) uniform states  $q_L$  and  $q_R$ . Approximate Riemann solvers have been proposed by Steger and Warming [St1], van Leer [Va1], Roe [Ro1], and Osher [Os1], [Os2]. A description of these upwind schemes and their properties can be found in the cited literature. For a consistent scheme,  $f^*(q, q) = f^*(q)$ , i.e., the

numerical flux function with equal arguments conforms with the genuine flux function in (1.1c). All these upwind flux functions are purely one-sided if all characteristics point in the same direction, i.e.,  $f^*(q_L, q_R) = f^*(q_L)$  if the flow of all information is from left to right.

**3.1.3. The multiple grid methods.** When a multiple grid technique is used to solve the system of nonlinear (differential) equations (1.7) or (1.8), we assume the existence of a nested set of grids. Usually this nesting is such that a set of  $2 \times 2$  cells in a fine mesh forms a single cell in the next coarser one. (No staggered grids!) The coarser grids are used to effect the acceleration of a basic iterative (time marching or relaxation) procedure on the finest grid.

Slightly generalizing equations (1.7) and (1.8) to

$$(1.9) \quad \frac{\partial}{\partial t} \mathbf{q}^h = N^h(\mathbf{q}^h) - r^h$$

and

$$(1.10) \quad N^h(\mathbf{q}^h) = r^h,$$

where  $r^h$  denotes a possible correction or forcing term, we can write the basic iterative procedure as

$$(1.11) \quad q^h \leftarrow J^h(q^h, r^h).$$

Generally, for a nonlinear equation this will be a nonlinear operation (e.g., a nonlinear Gauss–Seidel relaxation scheme).

The usual coarse grid acceleration algorithm is as follows: with an approximation  $q_{(k)}^h$  on the finest mesh, and some approximation  $q_{(0)}^{2h}$  on the next coarser one (e.g.,  $q_{(0)}^{2h} = I_h^{2h} q_{(k)}^h$ ), first an approximate solution  $q_{(1)}^{2h}$  is found for the coarse grid problem

$$(1.12) \quad N^{2h}(\mathbf{q}^{2h}) = N^{2h}(q_{(0)}^{2h}) - \hat{I}_h^{2h}(N^h(q_{(k)}^h) - r^h),$$

and then the value  $q_{(k)}^h$  is updated by

$$(1.13) \quad q_{(k+1)}^h = q_{(k)}^h + I_{2h}^h(q_{(1)}^{2h} - q_{(0)}^{2h}).$$

Notice that  $\hat{I}_h^{2h}$  is a restriction operator similar to  $I_h^{2h}$ ; the difference is that  $I_h^{2h}$  works on approximate solutions  $q^h$  (the state of the flow), whereas  $\hat{I}_h^{2h}$  works on residuals (rates of change of the flow). The difference is not only formal: in the simplest case  $I_h^{2h}$  takes the mean value of states in a set of cells, but  $\hat{I}_h^{2h}$  performs a summation of rates of change over a set of cells.

The combination of (1.12) and (1.13) is a *coarse grid correction* (CGC). The solution  $\mathbf{q}^{2h}$  of (1.12) can be approximated, e.g., by an (accelerated) iteration process on the  $2h$ -grid again. As for linear problems, by the recursive application of this idea we can form  $V$ -cycles or  $\mu$ -cycles.

We will see in §3.2 that the coarser grids sometimes play a role in the acceleration process that is different than the one we have just described [316], [474].

The *nonlinear multigrid cycle* (also called the *FAS-cycle*)

$$q^h \leftarrow FAS_\mu^h(q^h, r^h)$$

for the solution of (1.10) now consists of the following steps:

- Step 0:* Start with an approximate solution  $q^h$ .  
*Step 1:* Improve  $q^h$  by application of  $\nu_1$  nonlinear (pre-) relaxation iterations (1.11) to  $N^h(q^h) = r^h$ .  
*Step 2:* If the present grid is the coarsest, go to Step 4. Otherwise improve  $q^h$  by application of a coarse grid correction, where the approximation of (1.12) is effected by  $\mu$  FAS-cycles to this coarser grid problem; that is, compute

$$r^{2h} \leftarrow N^{2h}(q_{(0)}^{2h}) - \hat{I}_h^{2h}(N^h(q^h) - r^h),$$

and perform  $\mu$  times

$$q^{2h} \leftarrow FAS_\mu^{2h}(q^{2h}, r^{2h}).$$

- Step 3:*  $q^h \leftarrow q^h + I_{2h}^h(q^{2h} - q_{(0)}^{2h})$ .  
*Step 4:* Improve  $q^h$  by application of  $\nu_2$  nonlinear (post-) relaxation iterations to  $N^h(q^h) = r^h$ .

Again, the case with  $\mu = 1$  is called a *V-cycle*;  $\mu = 2$  yields a *W-cycle*. A *V-cycle* with  $\nu_1 + \nu_2 = 1$  is called a *sawtooth cycle*.

**3.2. Methods based on Lax–Wendroff type time stepping.** Ni [428] was among the first to apply an MG acceleration to the (isenthalpic) Euler equations. He uses the following time stepping procedure as a basic iteration. Starting with an initial state  $q_{(n)}^h$ , where the values  $q_{ij}^{(n)}$  are given at the grid points, he first computes the following quantities by means of a control volume centered integration method with fluxes interpolated from corner values:

$$\Delta q_{i+1/2, j+1/2} = -\frac{1}{2} \frac{\Delta t}{\Delta x} [(F_{i+1, j} - F_{i, j}) + (F_{i+1, j+1} - F_{i, j+1})]$$

$$(2.1) \quad -\frac{1}{2} \frac{\Delta t}{\Delta y} [(G_{i, j+1} - G_{i, j}) + (G_{i+1, j} - G_{i+1, j+1})],$$

$$F_{i, j} = F(q_{ij}^{(n)}), \quad \text{etc.}$$



These increments are then distributed over the mesh points using direction-weighted means (cell increments are distributed over mesh point values):

$$(2.2) \quad \Delta q_{ij} = \frac{1}{4} \sum_{l=\pm 1} \sum_{k=\pm 1} \left[ I - k \frac{\Delta t}{\Delta x} A_{i+k/2, j+l/2} - l \frac{\Delta t}{\Delta y} B_{i+k/2, j+l/2} \right] \Delta q_{i+k/2, j+l/2},$$

$$q_{ij}^{(n+1)} = q_{ij}^{(n)} + \Delta q_{ij}.$$

By way of the Jacobian matrices  $A$  and  $B$ , this distribution formula has an upwinding effect, but for transonic or supersonic cases an artificial damping is still necessary.

Symbolically, this time stepping process (2.1)–(2.2) is described as follows:

$$(2.3a) \quad \text{compute } \Delta q_{\text{cell}}^h,$$

with cell values  $\Delta q_{i+1/2, j+1/2} \approx -\Delta t \left( \int_{\partial \Omega_{i+1/2, j+1/2}} (f \cdot n_x + g \cdot n_y) ds \right) / (\Delta x \cdot \Delta y)$ ;

$$(2.3b) \quad q_{(n+1)}^h \leftarrow q_{(n)}^h + D^h \Delta q_{\text{cell}}^h.$$

The operator  $D^h$  is the distribution operator that transfers the cell centered corrections to the grid points by means of (2.2).

The coarse grid acceleration as introduced in [428] by Ni deviates from the canonical coarse grid scheme (1.12), (1.13). In [428] the coarse grid correction is obtained by first computing corrections at coarser cells,  $\Delta q_{\text{cell}}^{2h}$ . This can be done by restriction of  $\Delta q^h$  to the  $2h$ -grid. Then the corrections  $\Delta q_{\text{cell}}^{2h}$  are distributed to the coarser meshpoints as in (2.2), and the coarse grid correction is interpolated to the fine grid. Thus, here the coarse grid correction reads

$$(2.4a) \quad \Delta q_{\text{cell}}^{2h} \leftarrow I_h^{2h} \Delta q_{\text{cell}}^h,$$

$$(2.4b) \quad q_{(n+1)}^h \leftarrow q_{(n)}^h + I_{2h}^h D^{2h} \Delta q_{\text{cell}}^{2h},$$

where  $I_{2h}^h$  is a (bi-) linear interpolation operator. Since the coarse grid corrections are based on fine grid residuals, it is obvious that the possible convergence to a steady state yields a solution of the system (1.8).

In the same way the correction procedure can be repeated on progressively coarser grids. Therefore, in (2.4),  $2h$  should be replaced by  $2^k h$ . We notice that the corrections on the different levels may be made independently of each other. This makes it possible to compute all coarse grid corrections,  $k = 1, \dots, m$ , in parallel and to form the correction

$$q_{(n+1)}^h = q_{(n)}^h + \sum_{k=1}^m I_{2^k h}^h D^{2^k h} \Delta q_{\text{cell}}^{2^k h}$$

at once [541]. When optimal use of modern multi-processor computers is

made, it is also possible to perform both computations (2.3a) and (2.4) in parallel [320], [541].

We see that it is still possible to form different variants of the Ni-type multigrid Euler solver. First, any other Lax–Wendroff type time-marching procedure can be used for (2.3a). In [134], [314], [318] Johnson applies the popular MacCormack scheme. Further, in (2.4a) various restrictions,  $I_h^{2h}$ , can be used. Equation (2.4a) transfers the values of the fine grid corrections to a single value for each control volume in the coarser grid. Injection of the correction to the corresponding point of the coarse grid cell is often used [316], but weighted averages are also an obvious choice.

Heuristically, the coarse grid corrections in (2.4) have an accelerating effect because they may move disturbances of the steady state over the distance of many mesh cells in one time step. Apparently, the Lax–Wendroff schemes used in combination with this coarse grid correction must be sufficiently dissipative to reduce the high frequency disturbances present in the initial approximation or introduced by linear interpolation. One way to do this is to make a careful choice of  $\Delta t$ . Until now, no complete mathematical theory has been developed to explain or quantify the amount of acceleration clearly found in the use of this approach.

As an alternative to (2.2), where Jacobians are used to form the correction, Johnson [315] introduced a correction that is based on extrapolation (in time) of the computed fluxes.

**3.3. Methods based on semidiscretization and time stepping.** When only the solution of the steady state is to be computed, the time-accurate integration of the system of ordinary differential equations is wasteful. The convergence of (1.4) to steady state is slow. However, there may be several reasons to prefer time stepping methods, such as the desire to have a procedure that solves transient as well as steady state problems, coding convenience, or the restrictions imposed by the optimal use of vector computers. When no time accuracy is desired, many devices are known to accelerate the integration process (cf. [305]). For the solution of the Euler equations, these devices include: (i) local time stepping, which means that the step size in the integration process may differ over different parts of the domain  $\Omega^*$ ; (ii) enthalpy damping, where a priori knowledge about the behavior of the enthalpy over  $\Omega^*$  is used (e.g.,  $H$  constant over  $\Omega^*$ ); (iii) residual smoothing; and (iv) implicit residual averaging, which uses the fact that instability effects appear first for high frequencies, so that larger time steps are possible when the residual is smooth.

For all explicit integration methods, stability requirements set a limit on the size of the possible time steps (CFL limits). Implicit integration procedures can be unconditionally stable, but they require the solution of a nonlinear system at each individual time step.

An important code based on a time stepping method has been developed by Jameson, Schmidt and Turkel [305]. They use an explicit time stepping method of Runge–Kutta type. This *multistage time stepping procedure* is a specially adapted Runge–Kutta method, where the hyperbolic (convective) and the parabolic (dissipative) parts of  $N^h(q^h)$  are treated separately. The Runge–Kutta coefficients in the  $k$ -stage Runge–Kutta schemes ( $k = 3, 4$ ) are selected not only for their large stability bounds, but also with the aim of improving the damping of the high frequency modes. In the  $k$  stages of the Runge–Kutta process, the updating of the dissipative part is frozen at the first stage. This saves a substantial part of the computational effort.

The multigrid scheme used by Jameson [300] is an FAS sawtooth cycle with  $\nu_1 = 1$ . The restriction  $I_h^{2h}(\hat{I}_h^{2h})$  is defined by volume-weighted averaging of the states (summation of changes of states, respectively). The prolongation  $I_{2h}^h$  is defined by bilinear interpolation. The basic smoothing procedure is the “multistage time stepping scheme.” On the coarser grids the stability bounds for the time step, which are  $O(h)$ , allow larger time steps. On each grid the time step is varied locally to yield a fixed Courant number, and the same Courant number is used on all grids, so that progressively larger time steps are used after each transfer to a coarser grid. As for Ni’s method, the reasoning is that disturbances from the steady state will be more rapidly expelled from the domain  $\Omega^*$  by the larger time steps. The interpolation of corrections back to the fine grid introduces high frequency errors, which cannot be rapidly expelled. These errors should be locally damped. Hence, to obtain a fast rate of convergence, the time stepping process should rapidly damp the high frequency errors.

In [311] Jespersen announced an interesting theorem on the use of the MG process in combination with a time stepping procedure. This theorem asserts the following. Let  $I^h$  (resp.  $\hat{I}^h$ ) be defined as a restriction operator from the continuous state space (resp. space of rates of change) to its discrete equivalent on  $\Omega^h$ , and let  $I_h$  be a prolongation operator that interpolates states on  $\Omega^h$  to states on  $\Omega$ . Let  $N^h(\mathbf{q}^h) = 0$  be a space discretization of  $N(\mathbf{q}) = 0$  which is consistent, i.e.,

$$N^h(I^h(\mathbf{q})) - \hat{I}^h N(\mathbf{q}) = O(h),$$

and let the time stepping procedure be consistent in time, i.e.,

$$q_{(n+1)}^h = q_{(n)}^h + \Delta t_{(n)} [N^h(q_{(n)}^h) - r^h] + O((\Delta t_{(n)})^2).$$

If we consider the sawtooth algorithm, with  $\nu_1 = 1$ ,  $\nu_2 = 0$ ,  $\mu = 1$ , and if  $I_h$  and  $I^h$  satisfy an approximation property (i.e., for a smooth function  $q$  the prolongation and restriction in the state space are such that  $I_h I^h q - q = O(h)$ ), then the MG algorithm on  $m$  grids is a consistent, first order in time, discretization of (1.4) with time step  $\Delta t_{\text{tot}} = \sum_{j=1, \dots, m} \Delta t_j$ .

In a sense this theorem formalizes the heuristic reasoning that on coarser

grids the deviations from steady state can be expelled faster by the use of larger time steps. This may suggest that more, say  $\nu > 1$ , steps on the coarser grids would further improve the convergence. However, the theorem addresses consistency; stability is not considered. Hence, in the same paper [311] Jespersen shows by an example that convergence is lost when a large number of relaxations is made on the coarse grid. In fact, a strong stability condition of the form  $\Delta t/\Delta x \leq O(\nu^{-1})$  seems to appear.

**3.4. Fully implicit methods.** Most methods considered so far are based on the concept of integrating the equations (1.4) in time until a steady state is reached. If we are only interested in a possible solution of the steady state equation (1.5) and assume that this solution is unique, we may disregard the time-dependence completely. Further, assuming that a suitable space discretization takes into account the proper directions of dependence in  $\Omega^*$ , we can restrict ourselves simply to the solution of the nonlinear system (1.8) or

$$(1.10) \quad N^h(\mathbf{q}^h) = r^h.$$

Also, if the time-dependent system (1.9) is solved by means of an implicit time stepping method in order to circumvent the stability bounds on  $\Delta t$ , we have to solve systems (1.10) at each time step. Using these implicit solution methods and giving up time accuracy for (1.10) means that there is little or no difference between these time stepping procedures and (nonlinear) relaxation methods for (1.10).

If we start with the nonlinear system (1.10), two direct MG approaches can be used. We can either apply the nonlinear multiple grid algorithm (FAS) directly to the system (1.10), or we may apply linearization (Newton's method) and use the linear version of multiple grid (CS) for the solution of the resulting linear systems. Jespersen [310] gives an extensive recital of the (dis)advantages of both approaches. Both have been used with success for the Euler equations.

Linearization and CS have been used by Jespersen [309] and Mulder [423]; the nonlinear FAS procedure is used by Steger [528], Jespersen [309], and Hemker and Spekrijse [274], [275].

In all of these papers upwind discretizations have been used. In [309], [528] the Steger-Warming scheme is used; [423] uses the differentiable van Leer flux splitting method; [274], [275] use Osher's flux difference splitting. In [150] Dick also considers Roe's flux difference splitting for the 1-D Euler equations.

When Newton's method is applied for linearization, it may be difficult to start in the domain of contraction of the iteration. Therefore, Mulder [423] introduces the so-called Switched Evolution Relaxation (SER) scheme,

which is a chimera of a forward Euler time stepping and a Newton method:

$$(4.1) \quad \left[ \frac{1}{\Delta t} I - \frac{\partial}{\partial q} N^h(q_{(n)}^h) \right] (q_{(n+1)}^h - q_{(n)}^h) = N^h(q_{(n)}^h).$$

For  $\Delta t \rightarrow 0$ , this gives the simple time stepping procedure; for  $\Delta t \rightarrow \infty$ , (3.1) is equivalent to Newton's method. In the actual computation  $\Delta t$  varies, depending on the size of the residual, such that (3.1) is initially a time stepping procedure and becomes Newton's method in the final stages of the solution process.

In an FAS procedure, a natural way to obtain an initial estimate is of course to use full multigrid (FMG) [97]. The initial estimate is obtained by interpolation from the approximate solution on the coarser grid(s). For many problems this process gives very good results, even if one starts with rough approximations on every coarse grid.

**3.4.1. A nested sequence of Galerkin discretizations.** When (1.3) is discretized by a finite volume method, and if a conservative first order upwind (or a central difference) discretization is used as described in §3.1, it can be shown [275] that with a particularly simple restriction  $\hat{I}_h^{2h}$  and prolongation  $I_{2h}^h$ , the coarse discrete operator  $N^{2h}$  is a Galerkin approximation to the fine grid discretization  $N^h$ . With  $I_{2h}^h$  the piecewise constant interpolation over cells, and  $\hat{I}_h^{2h}$  the summation of the residual over fine mesh cells to form a residual on the corresponding coarse cell, the following relation holds:

$$(4.2) \quad N^{2h}(q^{2h}) = \hat{I}_h^{2h} N^h(I_{2h}^h q^{2h}).$$

This formula has an interesting implication for a coarse grid correction that is constructed by means of these operators. If the coarse grid correction (1.12), (1.13) transforms the approximation  $q_h$  into  $\bar{q}^h$ , the residual of  $\bar{q}^h$  satisfies

$$(4.3) \quad \hat{I}_h^{2h}[r^h - N^h(\bar{q}^h)] = \hat{I}_h^{2h}[(N^h q^h - N^h I_{2h}^h I_{2h}^h q^h) - (N^h \bar{q}^h - N^h I_{2h}^h I_{2h}^h \bar{q}^h)].$$

For a smooth operator  $N^h$ , this implies

$$\hat{I}_h^{2h}[r^h - N^h(\bar{q}^h)] = O(\|q^h - \bar{q}^h\|^2).$$

This means that the restriction of the residual mainly contains high frequency components. As is the case with common elliptic problems, it is the task of the relaxation method to efficiently damp these highly oscillating residuals.

**3.4.2. Relaxation methods.** Clearly, whether a sequence of Galerkin approximations is used or not, the important feature of a relaxation method in a multiple grid context (both CS and FAS) is its capability to damp the high frequency components in the error (or in the residual). Therefore, the difference scheme should be sufficiently dissipative as first order upwind schemes usually are. An advantage of these schemes over central differences is that this numerical dissipation is well defined and independent of an artificial parameter for the added dissipation necessary for the central difference schemes. The lack of differentiability of the numerical flux function may create a problem, but some differentiable flux functions are now available [528], [Os1], [Os2], [Va1].

Both in the linearized (CS) and in the nonlinear (FAS) application, well-known and simple relaxation procedures such as Gauss–Seidel (GS), symmetric Gauss–Seidel (SGS) and line Gauss–Seidel (LGS) are reported to work well when applied to the discrete Euler equations. (All of these relaxation methods are used in their “collective” version, i.e., the 3 or 4 variables corresponding to a single point or cell are relaxed simultaneously.) The smoothing behavior of these relaxations can be analyzed by local mode analysis. Here we should notice that the *smoothing factor*, as used for common elliptic problems, has no significant meaning for the Euler equation because we have to take into account characteristic (unstable) modes. A local mode analysis should follow more along the lines used for singularly perturbed elliptic problems (cf. e.g. [328]). Jespersen [309] has published some results in this regard. He shows that for a subsonic and supersonic case, SGS has a reasonably good smoothing behavior when applied to a first order scheme. Of course, the nonsymmetric GS relaxation is only effective if the direction of the relaxation sufficiently conforms with the direction of the characteristics. If we study plots of reduction factors of Fourier components (spectral radii, or norms for the error or residual amplification operator), e.g., when SGS is applied to the Euler equations, we see that two SGS sweeps are usually sufficient for a significant reduction of the high frequencies (Hemker, unpublished results). For second order schemes the smoothing rates are not satisfactory.

Van Leer and Mulder published a study [Va2] where several relaxation schemes (GS, LGS, ZEBRA, point Jacobi, line Jacobi, ADI, AF) were compared when applied to the linearized isenthalpic Euler equations.

**3.4.3. Higher order schemes.** When both first and second order upwind schemes are studied, the best MG performance is found for the first order discretizations. This can be explained by the fact that first order upwind schemes are more dissipative and hence more able to damp high frequencies. As first order schemes may not be accurate enough for practical computations and, moreover, have the unpleasant property of

TABLE 3.1  
Fully implicit multiple grid approaches.

	Discretization scheme	MG	Relaxation
Steger (1981)	Steger–Warming Finite differences	FAS	AF
Jespersen (1983)	Steger–Warming Finite differences	FAS/CS	SGS, GS
Mulder (1984)	van Leer Finite differences	CS	SGS
Hemker and Spekreijse (1985)	Osher Finite volumes	FAS Nested Galerkin	SGS, Damped Jacobi

smearing out skew discontinuities, second order schemes are highly desirable.

Beside the possibility of applying the MG acceleration directly to the second order scheme—with the unwanted effect of slowing the convergence rate—another possibility exists. Starting with a first approximation, we can improve the accuracy by the defect correction iteration [66], [271], [525]

$$(4.4) \quad N_1^h(q_{(n+1)}^h) = N_1^h(q_{(n)}^h) - N_2^h(q_{(n)}^h).$$

Here  $N_p^h$ ,  $p = 1, 2$ , denotes the  $p$ th order discretization. A theorem [225] has shown that for smooth solutions a single correction step (3.4) is sufficient to obtain the higher order of accuracy. Also, for solutions with discontinuities (where the formal order of convergence has no practical meaning), it is shown in [271] that one or a few steps (3.4) improve the accuracy of the solution significantly.

In Table 3.1 we summarize the several attempts to solve the steady Euler equations by an MG method with implicit relaxation. It is our opinion that the recent methods of this class are the most robust and efficient ones for solving the steady Euler equations. The development in the last few years has led to a significant improvement of the algorithms. However, the fully implicit methods have a rather complex structure and are not directly suited for vector computers. Furthermore, at the moment there is much less practical experience with these methods than, e.g., with Jameson's multistage time stepping procedure or the commonly used Beam–Warming [Be1] algorithm.

**Acknowledgments.** We are grateful to B. van Leer, whose comments on a draft of this paper led to several improvements. The investigations were supported in part by the Netherlands Technology Foundation.

## REFERENCES

- [Be1] R. M. BEAM AND R. F. WARMING, *An implicit finite-difference algorithm for hyperbolic systems in conservation-law form*, J. Comput. Phys., 22 (1976), pp. 87–110.
- [La1] P. D. LAX, *Shock waves and entropy*, in Contributions to Non-linear Functional Analysis, E. H. Zarantonello, ed., Academic Press, New York, 1971.
- [La2] ———, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, CBMS-NSF Regional Conference Series in Applied Mathematics 11, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1973.
- [Os1] S. OSHER, *Numerical solution of singular perturbation problems and hyperbolic systems of conservation laws*, in Analytical and Numerical Approaches to Asymptotic Problems in Analysis, O. Axelsson, L. S. Frank and A. van der Sluis, eds., Mathematics Studies 47, North-Holland, Amsterdam, 1981.
- [Os2] S. OSHER AND F. SOLOMON, *Upwind difference schemes for hyperbolic systems of conservation laws*, Math. Comp., 38 (1982), pp. 339–374.
- [Ro1] P. L. ROE, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comput. Phys., 43 (1981), pp. 357–372.
- [St1] J. L. STEGER AND R. F. WARMING, *Flux vector splitting of the inviscid gasdynamics equations with application to finite difference methods*, J. Comput. Phys., 40 (1981), pp. 263–293.
- [Va1] B. VAN LEER, *Flux-vector splitting for the Euler equations*, Proc. 8th Internat. Conf. on Numerical Methods in Fluid Dynamics, Aachen, June 1982, Lecture Notes in Physics 170, Springer-Verlag, New York–Berlin–Heidelberg.
- [Va2] B. VAN LEER AND W. A. MULDER, *Relaxation methods for hyperbolic conservation laws*, in Numerical Methods for Euler Equations of Fluid Dynamics, F. Angrand, A. Dervieux, J. A. Desideri and R. Glowinski, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1984.