JOURNAL OF COMPUTATIONAL PHYSICS **147,** 293–317 (1998)
ARTICLE NO. CP986056

Designing an Efficient Solution Strategy for Fluid Flows

II. Stable High-Order Central Finite Difference Schemes on Composite Adaptive Grids with Sharp Shock Resolution[1]

Margot Gerritsen* and Pelle Olsson†

**Department of Engineering Science, University of Auckland, Auckland, New Zealand;*
†Department of Scientific Computing, Uppsala University, 751 04 Uppsala, Sweden
E-mail: m.gerritsen@auckland.ac.nz, pelle@tdb.uu.se

Received September 25, 1997; revised June 24, 1998

A simple and efficient solution strategy is designed for fluid flows governed by the compressible Euler equations. It is constructed from a stable high-order central finite difference scheme on structured composite adaptive grids. This basic framework is suitable for solving smooth flows on complicated domains and is easily extendible with extra tools to handle specific flow problems. The stable high-order central difference scheme is mathematically formulated using a recently derived semi-discrete energy method for initial-boundary value problems. The high order of accuracy reduces the number of grid points required in smooth parts of the flow which leads to efficiency in both computational time and memory. A local grid adaptation technique is used to increase the grid density where required. Extra tools are developed for the sharp resolution of shocks. The grids are refined in the shock regions to retain accuracy. On the fine grids in these regions, an effective scalar artificial viscosity term is added to suppress spurious oscillations generated by the high-order central difference method. The location and orientation of shocks is determined by an easy-to-implement wavelet-based detection algorithm. The overhead of the composite adaptive grid method and detection algorithm is negligible compared to the computational kernel. The local grid adaptation with the high-order scheme is shown to increase computational efficiency. The resolution of shocks is sharp.

[1] This work has been sponsored by NSF Grant ASC 9318166-002 and by NASA under Contract NAS 2-13721.

1. INTRODUCTION

This paper discusses the ideas and tools that underlie a new general solution strategy for solving fluid flows governed by the compressible Euler equations. The emphasis in the design is on simplicity and efficiency. The basic method consists of a stable high-order central difference scheme implemented on structured composite adaptive grids. This combination is effective for solving smooth flows on complicated domains. Simple effective tools are built on top of this basic framework to handle more complicated problems. In this work, we developed a fast accurate wavelet-based detection algorithm and a cheap artificial viscosity term to show that we can effectively deal with shock waves, despite the notorious generation of spurious oscillations by central high-order schemes. This approach in designing a solution strategy differs from the current trend in academics to develop sophisticated schemes for specific flow situations as, for example, the essentially nonoscillatory (ENO) [35, 17] and total variation diminishing (TVD) [42, 11] schemes.

This paper builds on results presented in [16] in which the stable high-order central difference scheme, referred to as the SHOEC scheme, the artificial viscosity and the shock detection algorithm were introduced in one dimension. Here, we extend the method and tools to two dimensions and implement them on composite adaptive grids. The composite grid method, detection algorithm, and local grid adaptation approach are independent of the underlying numerical scheme and are therefore of general interest to the computational fluid dynamics community.

The scope of the solution strategy is not limited to the flow problems discussed in this paper. For example, we are investigating its use for the simulation of tidal flows. Turbulence models can be added to the scheme without affecting its basic properties. However, the method is not designed to deal with intricate shock interactions, moving boundaries, or very complex domains. Also the current framework is not designed to compute strong shock problems. It was our intention to develop a basic framework for smooth flows that can be easily and cheaply extended to deal with weak to moderate shocks as may occur in transonic flows.

1.1. Structured composite grid methods. Simplicity and efficiency were the two main criteria in choosing our numerical method. We kept future implementation on parallel architectures in mind. In general, structured grid codes have simple and transparent data structures, are straightforward to implement and lend themselves well for parallelization with languages such as high performance Fortran or auto-parallelizing compilers. The most commonly used structured grids are boundary-conforming curvilinear grids, in which the physical domain is mapped onto a square computational domain. However, for some complex geometries this mapping might not exist, or it might lead to low quality grids that create unacceptable errors in the numerical scheme. Unstructured grids, on the other hand, can be easily adapted to geometrical complexities. To overcome the limitations of structured grids several researchers started the development of methods based on nonboundary-conforming Cartesian grids; see a.o. [1, 4]. Such grids can handle complex configurations without the labor intensive case-by-case analysis that is normally required with body-fitted grids. But special treatment is required for the grid cells that are cut by the boundaries and it is difficult to maintain accuracy. In unstructured grids the grid points cannot be identified with coordinate lines. In two dimensions a grid is typically formed by triangular and quadrilateral cells, and in three dimensions tetrahedra, prisms, and pyramids are often used. The grids are body-fitted and very flexible. But unstructured grids require explicit bookkeeping of the grid connectivity which causes a larger memory overhead. Parallelization of these codes is harder and requires explicit message-passing to perform communication between processors. Much effort is being put into the development of parallel tools at present to try to catch up with the structured grid world [8]. Several strategies have been developed to combine the simplicity of the structured grids with the flexibility of the unstructured grids. Composite grid methods divide the flow domain into subregions each of which can accommodate a smooth curvilinear body-fitted structured grid. The underlying logical structure of composite grids allows the use of standard structured grid methods with little modification. The individual component grids can be patched or overlapping. In patched grids the different component grids share a common interface. Care must be taken so that overly skewed grid cells are not generated at the interfaces. Discontinuous or nonsmooth grid lines complicate the solution methods. In overlapping grids, communication of the solution between the component grids is done through interpolation, in our implementation this is a fourth-order Lagrangian interpolation, which offers greater geometric flexibility with smaller mesh generation times. Recent developments and grid generation packages are discussed by Pärt-Enander in [30], Zhu in [44], and Gerritsen in [15]. We chose finite differences on overlapping composite grids for their simplicity, efficiency, and suitability for parallelization. The composite grids offer sufficient flexibility for the applications of interest.

1.2. The SHOEC scheme. Ideally, a numerical method satisfies suitable stability and entropy (in)equalities and is locally well behaved. The construction of such schemes is hard and often limited to scalar equations in one dimension. Many schemes have been proposed with different properties based on different concepts. For example, the class of conservative methods was formulated to guarantee that the numerical solutions obtained are weak solutions so that the computed shock speeds are correct. Conservation is a sufficient, but not necessary, condition formulated for initial value problems (IVPs). As another example we mention the concept of total variation (TV) stability discussed in [20] which led to the construction of second-order conservative schemes that are stable for a wide class of one-dimensional scalar nonlinear IVPs. The MUSCL scheme developed by van Leer [42], and the PPM method proposed by Colella and Woodward [11] are based on this TV stability and are referred to as total variation diminishing (TVD) schemes or slope-limiter schemes. The TVD condition prevents spurious oscillations at shocks, but does not ensure that an entropy condition is satisfied. The essentially nonoscillatory (ENO) schemes designed by Harten, Osher, Engquist, and Chakravarthy [35, 17] were inspired by the slope-limiter techniques employed in the construction of TVD methods. They prevent the generation of spurious oscillations and are third or higher order accurate away from shocks and smooth extrema. They are not necessarily TV stable, nor can they in general be shown to satisfy an entropy condition, but they resolve shocks very sharply. The underlying theory is based on scalar one-dimensional IVPs. Recent developments are the complex fifth-order ENO schemes designed by Lindström [21] and the weighted ENO (WENO) schemes constructed by Shu [18]. All these schemes have desirable qualities but do not fit our criteria of simplicity and efficiency. Central difference schemes, on the other hand, do. Until recently, it was not known how to derive stability estimates for high-order central difference schemes. But advances by Kreiss, Scherer, Strand, and Olsson [19, 41, 26] led to the construction of high-order boundary difference operators that enabled the design of stable high-order schemes for linear systems. Olsson [25] extended these ideas to include nonlinear conservation laws. His results were built upon in [16] to design the stable high-order central difference scheme for the Euler equations. The derivation is based on symmetrization of the equations through entropy variables and a suitable splitting of the flux derivative vector. We refer to it as the split high-order entropy-correct (SHOEC) scheme. The SHOEC scheme satisfies an energy estimate. Entropy inequalities can be obtained if artificial viscosity is added near shocks. This is also desirable to suppress spurious oscillations generated by the high-order central scheme. We constructed a cheap scalar artificial viscosity term in [16] that led to sharply resolved shocks and retained the stability properties of the SHOEC scheme. Here, we extend it to two dimensions. We remark that it is not necessary to add artificial viscosity outside shock regions to stabilize the numerical method because of the inherent stability of the SHOEC scheme.

1.3. Local grid adaptation. In large scale problems it is far more memory and time efficient to concentrate grid points in regions where more accuracy is required while keeping the grid density low in regions where the flow is smooth. Often, the locations of these regions are not known a priori and they can change position in the flow domain. Thus, we use dynamic grid adaptation. Of course, a high grid density is needed in regions where numerical errors exceed a predefined tolerance. For linear problems, error estimation follows easily from stability estimates. For nonlinear equations, the numerical errors can be approximated after linearization if the flow is sufficiently smooth [44]. Recently Sjögreen [38] showed that all numerical schemes for systems will degrade to first-order accuracy at shocks, so accuracy can only be retained by refining the grid in the shock region. Error estimation techniques break down at discontinuities and therefore shock detection algorithms are needed to find the location of the shocks. In [16] a fast and accurate detection algorithm was developed based on a wavelet-analysis of the pressure grid function in one dimension. In two dimensions, it also gives information about the orientation of the shocks as is discussed in this paper. If the shock is sufficiently aligned with the grid, this information can be used to refine the grid only in the direction normal to the shock in which the flow variables change most rapidly. In the future, this information could be used to construct an aligned composite grid in the shock region to enhance the solution quality. Smearing of the shock on the coarse grids can further be avoided by adding artificial viscosity only on the finer grids surrounding the shock that were introduced to retain accuracy.

In summary, the solution strategy that we describe has the following building blocks:

- Split high-order entropy-correct finite difference scheme (Section 3)
- Overlapping composite adaptive grids for complicated domains (Section 4)
- Artificial viscosity to suppress spurious oscillations (Section 5)
- A detection algorithm to locate phenomena for grid refinement (Section 6).

We validate the components with an Euler test suite in Section 7. Conclusions and future directions are discussed in Section 8.

2. THE COMPRESSIBLE EULER EQUATIONS

The solution strategy and tools are tested on two-dimensional flow problems governed by the compressible Euler equations. We write them as a system of conservation laws

$$\begin{aligned} u_t + f_x + g_y = 0, \quad u,\ f, g \in \mathbb{R}^4, \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t > 0, \\ u(x, y, t = 0) = \phi(x, y). \end{aligned} \tag{2.1}$$

The state vector u and flux vectors f and g are given by

$$u = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{bmatrix}, \quad g = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix}.$$

The variables ρ, u, v, E, and p are the density, velocity in the x-direction, the velocity in the y-direction, the energy, and the pressure of the gas, respectively. The pressure is related to the velocities through the equation of state for a polytropic gas,

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right), \tag{2.2}$$

where $\gamma = 1.4$ is the ratio of specific heats c_p/c_v. At flow boundaries of the domain we prescribe data for the in-going characteristic variables ω_I by

$$\omega_I(\partial\Omega, t) = \psi(t).$$

At solid boundaries we require the velocity normal to the boundary to vanish. The flux vector $f(u)$ is a homogeneous function of order one so $f(\theta u) = \theta f(u)$. The Jacobian f_u is nonsymmetric.

The entropy S is given by

$$S = c_v \log(p\rho^{-\gamma}) + \text{const.}$$

In the remainder of this paper we will use a simplified scaled expression for the entropy of the form

$$S = \log(p\rho^{-\gamma}). \tag{2.3}$$

3. THE NUMERICAL SCHEME

We argued that a desirable numerical method has three main attributes: simplicity, high order of accuracy, and stability. Simplicity leads to efficiency. A high order of accuracy leads to savings in memory usage and computational time in smooth parts of the flow, again leading to efficiency. Stability ensures the numerical method is well-behaved and implies knowledge about the numerical errors that can be used in a grid-adaptive approach. In [16] we showed how to apply the results presented in [19, 41, 26, 25] to design a suitable high-order central numerical scheme for the Euler equations of gas dynamics in one dimension. If the flow is nonsmooth we add artificial viscosity in a neighborhood of the shock(s) which we show results in an entropy inequality. Here, we present the correct two-dimensional form of the SHOEC scheme. The complete derivation of the stability results for the (fully nonlinear) semi-discrete Euler equations in two space dimensions and general coordinates are given in [15]. The resulting semi-discrete equations are given in Eq. (3.4).

Studying this equation the reader may be disconcerted that what we are proposing is a scheme in nonconservative form. However, the scheme does satisfy a stability estimate and an entropy inequality in the presence of shocks. We realise that this is not a solid mathematical proof that the scheme yields a unique, physically correct solution (except in the case of scalar, convex conservation laws as shown by DiPerna [14]), but it is a desirable property. We would also like to mention that Iollo and Salas [34] recently showed how to obtain the correct Rankine–Hugoniot conditions from a primitive form of the Euler equations. We are currently investigating if a similar approach can be used to derive the Rankine-Hugoniot equations from the proposed SHOEC form. In [16] the shock speed computed by the SHOEC scheme in one-dimensional test problems was found to be correct. In practise, the scheme is implemented on composite adaptive grids. When shocks are present, we refine the grid and add artificial viscosity on the fine grid enclosing the shock. Thus, we solve a parabolic problem on the finest level which yields sharp resolution of shocks on the coarsest level. We believe that the proposed approach warrants further investigation in the transonic flow regions. In this paper we show that the scheme can handle weak shocks well. But, problems that are dominated by strong shocks probably warrant a different approach from the one we propose.

We remark that the stability proofs are valid for single uniform grids. Stability proofs for numerical methods on multi-dimensional composite grids are not available, but several authors have succeeded in deriving results in one space dimension; see, amongst others, [40, 39, 5].

3.1. The appropriate form in two dimensions. As in the one-dimensional case, we symmetrize the Euler equations by transformation to the so-called entropy variables $w(u)$, and apply a splitting to the fluxes to obtain

$$u_t = -\frac{1}{\beta+1}((\tilde{f}_w w)_x + (\tilde{g}_w w)_y + \tilde{f}_w w_x + \tilde{g}_w w_y). \tag{3.1}$$

As discussed in [16] the parameter β is given by $\beta = (\alpha+\gamma)/(1-\gamma)$, where $\alpha > 0$ or $\alpha < -\gamma$. The entropy variables $w(u)$, the fluxes $\tilde{f}$ and $\tilde{g}$, and the Jacobians $\tilde{f}_w$ and $\tilde{g}_w$ are given in Appendix A. We assume that the mapping between physical coordinates (x, y) and the computational coordinates $(r, s) \in (0, 1) \times (0, 1)$ is one-to-one and that both the mapping and its inverse are continuously differentiable. The physical and computational coordinates satisfy the relations

$$\begin{aligned} x_r &= J s_y, & y_r &= -J s_x, \\ x_s &= -J r_y, & y_s &= J r_x, \end{aligned} \qquad J = \det\begin{bmatrix} x_r & x_s \\ y_r & y_s \end{bmatrix}.$$

These equations imply that

$$\frac{\partial}{\partial r}(J(r_x + r_y)) + \frac{\partial}{\partial s}(J(s_x + s_y)) = 0. \tag{3.2}$$

The Jacobian of the transformation and its determinant J are, of course, independent of time. We further assume that $|J| > 0$ always. In terms of the computational coordinates, Eq. (3.1) equals

$$\begin{aligned} u_t = &-\frac{1}{\beta+1}(r_x(\tilde{f}_w w)_r + s_x(\tilde{f}_w w)_s + r_y(\tilde{g}_w w)_r + s_y(\tilde{g}_w w)_s) \\ &- \frac{1}{\beta+1}(r_x \tilde{f}_w w_r + s_x \tilde{f}_w w_s + r_y \tilde{g}_w w_r + s_y \tilde{g}_w w_s). \end{aligned} \tag{3.3}$$

We use Eq. (3.2) and the homogeneity relations to rewrite Eq. (3.3) as

$$\begin{aligned}(Ju)_t &= -\frac{\beta}{\beta+1}((Jr_x\tilde{f} + Jr_y\tilde{g})_r + (Js_x\tilde{f} + Js_y\tilde{g})_s)\\ &\quad - \frac{1}{\beta+1}(Jr_x\tilde{f}_w w_r + Jr_y\tilde{g}_w w_r + Js_x\tilde{f}_w w_s + Js_y\tilde{g}_w w_s),\end{aligned}$$

and we discretize it as

$$\begin{aligned}\boldsymbol{u}_t &= -\frac{\beta}{J(\beta+1)}\{D_r(Jr_x\tilde{\boldsymbol{f}} + Jr_y\tilde{\boldsymbol{g}}) + D_s(Js_x\tilde{\boldsymbol{f}} + Js_y\tilde{\boldsymbol{g}})\}\\ &\quad - \frac{1}{\beta+1}\{(r_x\tilde{f}_w + r_y\tilde{g}_w)D_r\boldsymbol{w} + (s_x\tilde{f}_w + s_y\tilde{g}_w)D_s\boldsymbol{w}\}.\end{aligned} \tag{3.4}$$

Both D_r and D_s are assumed to satisfy the equality

$$(\boldsymbol{u}, D\boldsymbol{v})_h = \boldsymbol{u}_n^{\mathrm{T}}\boldsymbol{v}_n - \boldsymbol{u}_0^{\mathrm{T}}\boldsymbol{v}_0 - (D\boldsymbol{u}, \boldsymbol{v})_h, \tag{3.5}$$

in respectively the r and s-directions. This is the summation-by-parts principle that was shown in [16] to be an essential ingredient in deriving the stability results. The scalar products used in [15, 27] to prove stability are defined as

$$\langle \boldsymbol{u}, \boldsymbol{v}\rangle_h \equiv (\boldsymbol{u}, J\boldsymbol{v})_h = (J\boldsymbol{u}, \boldsymbol{v})_h, \tag{3.6}$$

where

$$(\boldsymbol{u}, \boldsymbol{v})_h = \Delta r\,\Delta s \sum_{i,j=0}^{n} \sigma_i\sigma_j u_{ij}^{\mathrm{T}} v_{ij} \equiv h\boldsymbol{u}^{\mathrm{T}}\Sigma\boldsymbol{v}.$$

The term u_{ij} represents the solution in the grid point $x_{ij} = (i\Delta r, j\Delta s)$. In (3.6) each point is scaled with the cell volume. This scalar product is a natural choice as the analytic scalar product is equal to $(u, v) = \int_\Omega u^{\mathrm{T}} v\, d\Omega = \int_{(0,1)\times(0,1)} u^{\mathrm{T}}(x(r,s), y(r,s))v(x(r,s), y(r,s))J\,dr\,ds$. As shown in [15] the semi-discrete equations (3.4) allow the derivation of a multidimensional nonlinear semi-discrete energy estimate. In the experiments performed in this paper we used discretization operators with an overall accuracy four. It is sixth order in the interior and third order at the boundary. Its coefficients are given in [16].

The equations (3.4) are solved in time by the third-order total variation diminishing Runge–Kutta scheme formulated in [36]. Although our spatial accuracy is generally fourth-order, we elected this scheme for its simplicity and stability properties. For a system of ODEs $\boldsymbol{u}_t = L(\boldsymbol{u}(\boldsymbol{w}))$, as in (3.4), the scheme computes the solution $\boldsymbol{u}^{n+1}$ at the $(n+1)$th time step according to

$$\begin{aligned}\tilde{\boldsymbol{u}}_1 &= \boldsymbol{u}^n + \Delta t L(\boldsymbol{u}^n),\\ \tilde{\boldsymbol{u}}_2 &= 3/4\boldsymbol{u}^n + 1/4\tilde{\boldsymbol{u}}_1 + 1/4\Delta t L(\tilde{\boldsymbol{u}}_1),\\ \boldsymbol{u}^{n+1} &= 1/3\boldsymbol{u}^n + 2/3\tilde{\boldsymbol{u}}_2 + 2/3\Delta t L(\tilde{\boldsymbol{u}}_2),\end{aligned} \tag{3.7}$$

with time-step Δt and intermediate variables $\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{u}}_2$.

The boundary projection is applied at the end of the complete Runge–Kutta cycle to preserve the accuracy of the RK method in accordance with the recommendation by Carpenter, Gottlieb, Abarbanel, and Don [9].

4. OVERLAPPING COMPOSITE ADAPTIVE GRIDS

The composite adaptive grid (CAG) method that we use is based on the CAG method proposed by Zhu in [44]. The composite grid is constructed with the extendible composite overlapping grid-generator XCOG developed by Petersson [31]. XCOG is a basic and efficient two-dimensional grid generator. The code is well structured so that the addition of modules developed by the user and coupling of XCOG to flow solvers is straightforward. A three-dimensional version is developed under the name CHALMESH [32] at Chalmers' Center for Marine Research and Technology, Sweden.

4.1. Overlapping composite grids. A composite grid is constructed from two or more component grids that are either Cartesian or curvilinear. We start with the construction of (small) body-fitted grids surrounding curved boundaries and then cover as much as possible of the remainder of the flow domain with a Cartesian grid to minimize the costs of the grid generation and the numerical computations. A simple example is shown in Fig. 1. The subdivision of the flow domain into several different component grids can be used to other advantages. It leads to natural domain decomposition methods for parallelization, for example. Also, different equations can be modeled on different grids. An example of the latter is the use of full Navier–Stokes on boundary grids and Euler on interior grids in high Reynolds number flows [30].

The grid points in each component grid can be assigned to one of two classes: *interior* points and *overlap* points. In Fig. 1 the overlap points are indicated by squares in the Cartesian grid G_1 and circles in the curvilinear grid G_2. We remark that this example corresponds to a central discretization stencil of width three and that wider stencils will require more overlap points. The two grid-point classes differ in the way their solution values are updated. The values in the interior points are determined by the difference method. Because the interior points of the component grids cover the complete flow domain (with a small overlay) a new solution is computed everywhere after the time step is completed. The values in the overlap points are computed by interpolation of the new solution values in a set of interior points of the overlapped grid (or donor grid). Recently much effort has been put in the design of interpolation schemes that are conservative (see, e.g., Berger [3] and Pärt and Sjögreen [29]). It does not pay to use these expensive schemes in the SHOEC scheme because the SHOEC scheme itself is in nonconservative form. Instead we use

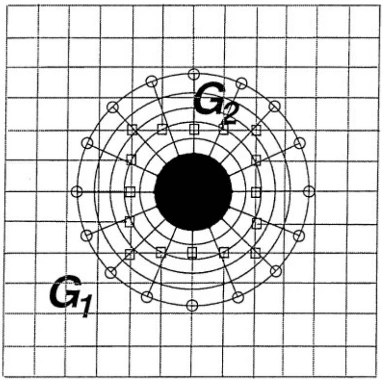


FIG. 1. Composite grids around a cylinder for a difference stencil with a width of three.

Lagrangian interpolation of an order consistent with the order of the overall scheme. Thus far we have not allowed shocks to travel through the overlap regions. We expect that the artificial viscosity added to our scheme around shocks (Section 5) will suppress oscillations that may be introduced by the high-order interpolation at the overlap boundaries and that automatic grid refinement in the shock regions will keep the resolution sharp.

So far, the effect of the overlap on stability has not been studied extensively. The most classical and perhaps most general result is by Starius in [40, 39]. He investigated hyperbolic equations in one dimension using the Lax–Wendroff method. His results indicated that the stability depends on the physical width of the overlap and that, in general, this width should be independent of the mesh size. Accordingly, we keep the width of the overlap constant when refining which means that the number of overlap points grows as the grid size decreases.

4.2. Local grid adaptation. The purpose of grid adaptation is to restrict high grid density to regions in which an increased numerical accuracy is needed, thus allowing coarse grids elsewhere. We use two complementary tools to detect these regions during simulation:

1. *Numerical error estimation based on the stability estimates.* For linear (systems of) equations error estimates follow immediately from the stability estimates of the semi-discrete solution on one grid. Typically, let v_h be the numerical solution to the linear problem

$$\begin{aligned} u_t &= Lu, && \text{on } \Omega \times [0, T], \\ u(x, 0) &= u_0(x), \\ u &= g(t), && \text{on } \partial\Omega. \end{aligned}$$

If v_h satisfies a stability estimate of the form

$$\|v_h(T)\|^2_{\Omega_h} \leq Ke^{\alpha T}\big(\|u_{0h}\|^2_{\Omega_h} + \|g\|^2_{\partial\Omega_h \times [0,T]}\big),$$

then the error $e_h \equiv u_h - v_h$, where u_h is the projection of the analytic solution u on the discrete domain Ω_h, will satisfy an estimate of the form

$$\|e_h(T)\|^2_{\Omega_h} \leq Ke^{\alpha T}\left(\|\tau\|^2_{\Omega_h \times [0,T]} + \|\tau_{\text{boundary}}\|^2_{\partial\Omega_h \times [0,T]}\right).$$

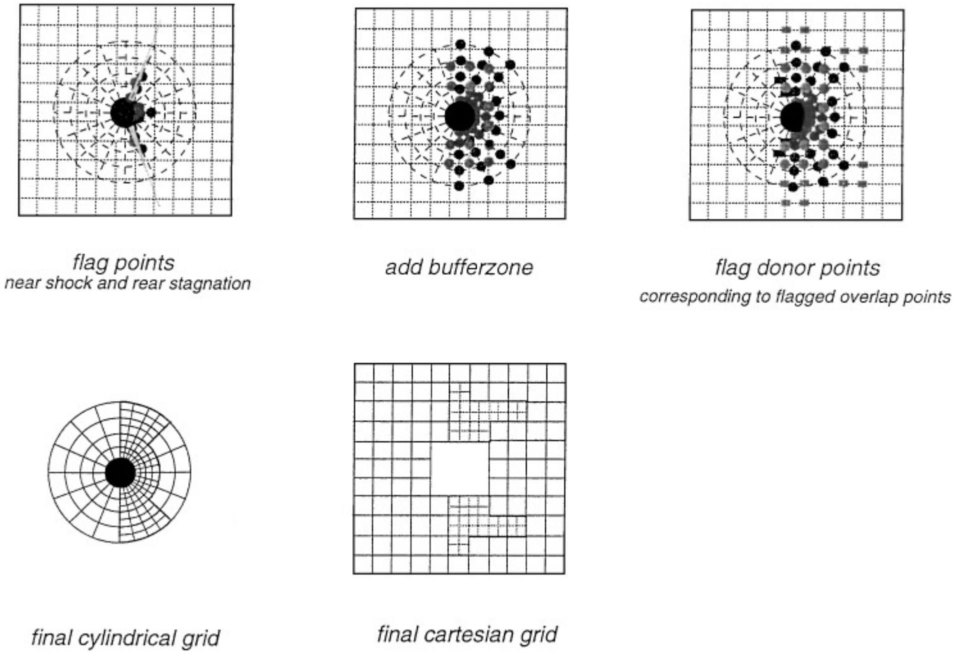
Here, τ is the truncation error. For nonlinear equations we can derive an error estimate after linearization. Zhu demonstrated the validity of this approach with numerical tests on the shallow water equations [44]. The error estimates can be generalized for composite grids as shown in the same work for low-order methods. We are currently investigating its extension to high-order methods.

2. *Shock detection based on a wavelet analysis of the pressure grid function.* Error estimation alone does not suffice. Recently, Sjögreen showed that the accuracy at a shock cannot exceed first-order on uniform grids, independent of the numerical scheme used. The low accuracy will gradually pollute the solution away from the shock as the solution is stepped forward in time. To retain accuracy, the grid has to be refined in the shock regions. We remark that formal accuracy may also be preserved using shock fitting, or subcell resolution. Because numerical error estimation is based on an approximation of the truncation error and presumes that the solution is smooth, the analysis will break down at discontinuities. A separate detection algorithm is needed to locate shocks. We proposed a wavelet-based detection algorithm in [16] which is extended to two dimensions in Section 6.

Grid refinement is achieved by the injection of extra grid points in the flagged regions, thus creating an overlaying fine grid. If the percentage of flagged points in a grid exceeds a certain fraction, the complete grid is refined. The ratio r between coarse and fine grid size is kept constant and equal to 4 which has been found to work well for hyperbolic equations [2]. The solution is stepped forward in time on the complete coarse grid. Then, solutions on the overlaying fine grids are computed for the same time, requiring r times as many time steps if the time step is controlled by a CFL condition. The boundary conditions for the boundaries of the fine grid that are located in the interior of the coarse grids are determined by interpolation in space and, if needed, in time of coarse grid values. At the end of the cycle the coarse grid solutions are updated from the fine grid solutions by injection. The refinement process is repeated recursively so that a series of increasingly finer grids may be created.

The size of the refined areas is kept to a minimum if the regridding is done at each step, but this is prohibited by the regridding costs. Therefore, we create a *buffer zone* of a few grid points wide around the flagged points and include it in the regridding. The addition of the buffer zone will keep moving features inside fine grids longer and takes care of possible error growth directly outside a flagged region before the next regridding takes place. If moving shocks or interfaces are detected, we approximate their speed and increase the buffer zone in the direction of propagation. We remark that if overlap points are flagged during the above process, their donor points are flagged as well to ensure sufficient accuracy in the interpolation.

A (oversimplified) graphical illustration of the grid adaptation process is given in Fig. 2. Here, we show a likely flagging of grid points in the simulation of transonic flows past a cylinder. Two shocks are formed on the lee side of the cylinder that are detected by the



flag points
near shock and rear stagnation

add bufferzone

flag donor points
corresponding to flagged overlap points

final cylindrical grid

final cartesian grid

FIG. 2. Possible flagging round a cylinder when modeling transonic flow.

shock detection algorithm. Points in the wake of the cylinder may be flagged by the error estimation method. A buffer zone is included and donor points are flagged if necessary. As shown, the resulting set of fine grid points is not usually contiguous, but gaps and holes and ragged fine grid boundaries may be present. We refer to such grids as stair-step grids for obvious reasons. In most composite adaptive grid methods the fine grid is decomposed into a set of rectangles. This simplifies the required data structures. But it is hard to find an efficient way to cover the flagged regions with rectangular patches. Instead, we use one data structure for each complete fine stair step grid that accommodates the grid irregularities. The required data structures are more complicated, but they can be created easily in C, C++, or Fortran 90. Stair-step grids are natural in composite grid methods as the coarse component grids have stair-step overlap boundaries. It is therefore easy to deal with refinements in the overlap regions. Stair-step grids were used by Blom, Trompert, and Verwer in their Fortran 77 VLUGR codes [7, 6]. However, Fortran 77 does not lend itself well to the required data structures.

5. ARTIFICIAL VISCOSITY

Artificial viscosity is needed at shocks to suppress spurious oscillations generated by the central high-order difference scheme. Ideally, artificial viscosity terms should be constructed such that the stability estimates are not destroyed, shocks are resolved sharply, and the artificial viscosity terms can be computed cheaply. In [16] we designed a cheap but effective artificial viscosity term in one dimension that satisfies these criteria. The artificial viscosity is introduced on a (sequence of) refined grid(s) around the shock, but not on the coarse grid. It is of the form

$$hQ\boldsymbol{w} = hD_{+}\epsilon u_w D_{-}\boldsymbol{w}, \tag{5.1}$$

where

$$(D_{+}\boldsymbol{u})_j = \frac{1}{h}(u_{j+1} - u_j), \quad (D_{-}\boldsymbol{u})_j = \frac{1}{h}(u_j - u_{j-1}),$$

and

$$\epsilon = \begin{cases} \frac{1}{2}\min_i |\lambda_i|, & s = 0, \\ \frac{1}{2}|\lambda_2 - s|, & s \neq 0. \end{cases} \tag{5.2}$$

Here, s is the shock speed and $\lambda_1 = u - c$, $\lambda_2 = u$, and $\lambda_3 = u + c$ are the characteristic speeds of the flow. They are evaluated at the Roe-averaged state $\tilde{u}$, determined by the upstream shock state u_l and the average $u_m = (u_l + u_r)/2$ of u_l and the downstream shock state u_r. The shock speed s for a k-shock is computed with $s \approx (\lambda_k(u_l) + \lambda_k(u_r))/2$. The Jacobian u_w is given in Appendix A. The term (5.1) is a consistent discretization of the diffusion term $(Eu_x)_x = (Eu_w w_x)_x$, scaled by the grid size h, which preserves the stability estimates if $E = 0$ at the boundary [16]. The addition of this artificial viscosity term naturally reduces the SHOEC scheme to first-order. This is acceptable because Sjögreen [38] found that any scheme for systems of conservation laws reduces to first-order accuracy at shocks. High-order of accuracy of a scheme can only be retained if the grid is refined in the shock region as we do. Because we only use artificial viscosity on the locally refined grids excessive smearing does not occur at the coarse grid scale and thus the smallness of the artificial viscosity is not that crucial. In fact, we can use as large an artificial viscosity as we want, as long as we keep the refined grids on which it is applied sufficiently fine.

5.1. The artificial viscosity term in two dimensions. In two dimensions we include an artificial viscosity term analogous to (5.1) for each computational coordinate direction in which it is needed. That is,

$$|J|\boldsymbol{u}_t = L(\boldsymbol{u}) + |J|(Q_r\boldsymbol{w} + Q_s\boldsymbol{w}), \tag{5.3}$$

where $L(\boldsymbol{u})$ is given by the right-hand side of (3.4), and

$$Q_r\boldsymbol{w} = hD_+^r \tilde{E}u_w D_-^r\, \boldsymbol{w}$$

with a similar expression for the s-direction. Here $h = \Delta r\,\Delta s$ is the cell area. The viscosity is scalar, i.e., $\tilde{E} = \tilde{\epsilon} I$. We remark that the viscosities $\tilde{\epsilon}$ as defined in (5.4) through (5.6) contain an extra Δr- or Δs-term in the denominator. If the grid is sufficiently aligned with a shock, artificial viscosity is only needed in the coordinate direction normal to the shock. Without proper alignment we resort to adding viscosity in both directions. Care must be taken to properly extend the scalar artificial viscosity (5.2). In contrast to the one-dimensional case, there are an infinite number of characteristic propagation directions, given by wave vectors $\boldsymbol{\kappa}$. The corresponding eigenvalues $\lambda_{\boldsymbol{\kappa}}$ are given by

$$\lambda_{\boldsymbol{\kappa}} = \{\mathbf{1}_{\boldsymbol{\kappa}} \cdot \overline{v}, \mathbf{1}_{\boldsymbol{\kappa}} \cdot \overline{v}, \mathbf{1}_{\boldsymbol{\kappa}} \cdot \overline{v} - c, \mathbf{1}_{\boldsymbol{\kappa}} \cdot \overline{v} + c,\},$$

where $\overline{v}$ is the velocity vector and $\mathbf{1}_{\boldsymbol{\kappa}}$ is the unit vector in the direction of $\boldsymbol{\kappa}$. Since we are interested in information propagating in the direction of the computational coordinates r and s, logical choices are $\boldsymbol{\kappa} = \nabla r = [\partial r/\partial x, \partial r/\partial y]^{\mathrm{T}}$ and $\boldsymbol{\kappa} = \nabla s = [\partial s/\partial x, \partial s/\partial y]^{\mathrm{T}}$. We propose the use of

$$\tilde{\epsilon} = \begin{cases} \frac{1}{2\Delta r}\min_i |\lambda_{\nabla r,i}|, & s = 0, \\ \frac{1}{2\Delta r}|\lambda_{\nabla r,2} - s|, & s \neq 0, \end{cases} \tag{5.4}$$

if the shock is aligned with the s-direction, and

$$\tilde{\epsilon} = \begin{cases} \frac{1}{2\Delta s}\min_i |\lambda_{\nabla s,i}|, & s = 0, \\ \frac{1}{2\Delta s}|\lambda_{\nabla s,2} - s|, & s \neq 0, \end{cases} \tag{5.5}$$

if the shock is aligned with the r-direction. If the shock is not aligned with the grid we take the average

$$\tilde{\epsilon} = \begin{cases} \frac{1}{4\Delta r}\min_i |\lambda_{\nabla r,i}| + \frac{1}{4\Delta s}\min_i |\lambda_{\nabla s,i}|, & s = 0, \\ \frac{1}{4\Delta r}|\lambda_{\nabla r,2} - s| + \frac{1}{4\Delta s}|\lambda_{\nabla s,2-s}|, & s \neq 0. \end{cases} \tag{5.6}$$

6. DETECTING PHENOMENA WITH WAVELETS

In the previous sections we motivated the need for an accurate and efficient detection algorithm to locate sharp interfaces and shocks. To retain the accuracy of the numerical scheme we need to refine the grid locally around shocks. Accurate detection of the shock location will minimize the refinement costs. We are interested in the position of shocks or interfaces as well as their orientation. If the grid is sufficiently aligned with a shock, refinement in the direction normal to the shock will suffice. Also, addition of artificial viscosity can be restricted to this computational direction. In future applications the information about the orientation of the shock may be used to reconstruct the grid in the shock region. In [16] we presented an efficient and accurate detection algorithm in one dimension based on a multiscale wavelet analysis of a grid function, i.e. the pressure grid function. The algorithm returns accurate information about the shock location and can also easily detect spurious oscillations. In this section we show how the one-dimensional wavelet detection can be extended to two dimensions.

In the last few years a number of researchers have investigated the use of wavelets in computational fluid dynamics (see, e.g. [33, 43]). In the majority of the proposed methods the partial differential equations are solved in wavelet space; the grid functions are expanded in terms of a wavelet basis and appropriate operators, e.g. differentiation operators, are constructed in the wavelet space, generally in the form of Galerkin or collocation methods. Data compression in wavelet space, achieved by discarding all wavelet coefficients below a certain threshold value, corresponds to grid adaptation in physical space. The wavelet-based schemes produce high grid density in regions where the flow variables change rapidly and low density elsewhere, because slow variations correspond to small wavelet coefficients. Complex geometries, arbitary meshes, nonlinearities, and nonperiodic boundary conditions are, however, difficult, if not as yet impossible, to handle. In recent work, Walden [43] used filterbanks instead of the standard orthonormal wavelets in combination with finite difference methods. His filterbank method is computationally more efficient and simplifies the treatment of nonlinearities and general boundary conditions. However, restrictions on the computational domain remain.

Our approach is different; we use wavelet analysis only at the postprocessing stage, thus making it fully independent of the numerical method used and avoiding any problems associated with irregular computational domains, such as stair-step grids, boundary conditions, or nonlinearities. We are also primarily interested in detecting discontinuities and local oscillations. If a more accurate error estimation is required we can use the error estimation technique explained in Section 4.2. With this approach we are able to use well-established and efficient wavelet detection algorithms developed for edge and noise detection in signal analysis as a basis for our detection algorithm. The wavelets used are nonorthonormal spline wavelets with symmetric, compact support designed by Mallat [22–24]. The shock detection ability of these wavelets in finite discrete domains is shown to exceed the detection ability of popular Daubechies wavelets [12].

In the remainder of this section we very succinctly present the one-dimensional theory first and then show how the wavelet algorithm is extended to two dimensions. Detailed discussion can be found in [15, 16].

In one dimension a function f is convolved with dilations and translations $\psi((x-b)/a)$ of a mother wavelet $\psi(x)$. The parameter b is referred to as the *center* of the wavelet $\psi_{a,b}$

and a represents its *scale*. The wavelet coefficients

$$\langle f, \psi_{a,b}\rangle \equiv \frac{1}{a}\int f(x)\psi\left(\frac{x-b}{a}\right)dx$$

give information on the frequency content of $f(x)$ in the neighborhood of $x=b$. We discretize a as $a_m = 2^m$, $m \in \mathcal{Z}$, where m represents the level. The grid size h is normalized to 1 (corresponding to level $m=0$). At each level m the wavelet coefficients are computed at all grid points $x_i = i$. The mother wavelet $\psi(x)$ used is a quadratic anti-symmetric spline with compact support $[-1, 1]$ given by

$$\psi(x) = \begin{cases} 2(x+1)^2, & -1 \le x < -1/2, \\ -4x(1+x) - 2x^2, & -1/2 \le x < 0, \\ -4x(1-x) + 2x^2, & 0 \le x < 1/2, \\ -2(x-1)^2, & 1/2 \le x < 1. \end{cases}$$

As discussed in [16] it is the derivative of a compact smoothing function $\theta(x)$. In this same reference it is shown that each wavelet coefficient $\langle f, \psi_{a,b}\rangle$ is equal to the first derivative of the function $f(x)$ smoothed at the scale a by the function $\theta_{a,b} = a^{-1}\theta((x-b)/a)$. At a shock, the first derivative will have a local maximum. The one-dimensional detection algorithm, therefore, first collects the local modulus maxima of the wavelet coefficients at all scales. Furthermore, modulus maxima that correspond to shocks do not vary from one scale to the next. This is used to pick out the desired wavelet coefficients and, thus, the shock locations.

In two dimensions we use two wavelets $\psi^1(x, y)$ and $\psi^2(x, y)$ given by

$$\psi^1(x, y) = \frac{\partial\tilde{\theta}(x, y)}{\partial x},$$
$$\psi^2(x, y) = \frac{\partial\tilde{\theta}(x, y)}{\partial y},$$

where $\tilde{\theta}(x, y) = \theta(x)\theta(y)$.

The wavelet transform of a function $f(x, y)$ has two components

$$\begin{aligned} \left\langle f, \psi^1_{a,b,c}\right\rangle &\equiv \left\langle f, a^{-1}\psi^1\left(\frac{x-b}{a}, \frac{y-c}{a}\right)\right\rangle, \\ \left\langle f, \psi^2_{a,b,c}\right\rangle &\equiv \left\langle f, a^{-1}\psi^2\left(\frac{x-b}{a}, \frac{y-c}{a}\right)\right\rangle. \end{aligned} \tag{6.1}$$

Again, a represents the scale and b and c are centers of the wavelet in, respectively, the x- and y-directions. As in the one-dimensional case, we discretize a as $a = 2^m$ and compute wavelet coefficients at all grid points $x_{i,j} = (i, j)$.

If, as in (6.1), we use the same scale a in both x- and y-directions in two dimensions, it can be shown that

$$\begin{aligned} \begin{bmatrix} \left\langle f, \psi^1_{a,b,c}\right\rangle \\ \left\langle f, \psi^2_{a,b,c}\right\rangle \end{bmatrix} &= -a \begin{bmatrix} \frac{\partial}{\partial b}\langle f, \theta_{a,b,c}\rangle \\ \frac{\partial}{\partial c}\langle f, \theta_{a,b,c}\rangle \end{bmatrix} \\ &= -a\nabla\langle f, \theta_{a,b,c}\rangle. \end{aligned}$$

Consequently, the components of the two-dimensional wavelet transform at $(x = b, y = c)$ are proportional to the components of the gradient vector of $f(x, y)$ smoothed at the scale a. The direction of the gradient vector indicates the direction in which $f(x, y)$ has the largest variation. Shocks can therefore be found perpendicular to gradients that are local modulus maxima. So again we search for local modulus maxima first. The modulus $M_{a,b,c}$ of the gradient vector is defined as

$$M_{a,b,c} = \sqrt{\left|\left\langle f, \psi^1_{a,b,c}\right\rangle\right|^2 + \left|\left\langle f, \psi^2_{a,b,c}\right\rangle\right|^2}.$$

We investigate the behaviour of the modulus maxima across scales as in the one-dimensional case. Finally, to recover shock curves we chain adjacent local maxima if their respective position is perpendicular to the gradient direction.

In [16] we showed that the wavelet coefficients in one dimension can be computed efficiently using two discrete and finite filters G and H. In two dimensions this fast wavelet transformation (FWT) is also possible. We can express the two-dimensional FWT in terms of the one-dimensional filter operations as

$$\begin{aligned} F_m &= P_{m-1} * (H_{m-1}, H_{m-1}), \\ W^1_m &= \frac{1}{\lambda_m} P_{m-1} * (G_{m-1}, D), \\ W^2_m &= \frac{1}{\lambda_m} P_{m-1} * (D, G_{m-1}). \end{aligned}$$

Here, $P_{m-1} * (G_{m-1}, D)$ denotes the convolution of the rows of the two-dimensional signal P_{m-1} with the one-dimensional filter G_{m-1} and the columns of the signal with the one-dimensional filter D, and similarly for the other expressions. The filter D is the Dirac filter with impulse response 1 at 0 and 0 everywhere else.

7. RESULTS

We tested the solution strategy preliminary on an Euler test suite. Here we present the results for a traveling isentropic vortex and transonic flow past a cylinder. The first problem is used to show the effectiveness of high-order methods in smooth flows. The second example shows the effectiveness of the combination of detection algorithm and artificial viscosity. In the experiments we use a sixth-order SHOEC scheme in the interior of the flow and the appropriate third-order one-sided stencils at nonperiodic boundaries. Interpolation in the overlap of composite grids is fourth-order Lagrangian.

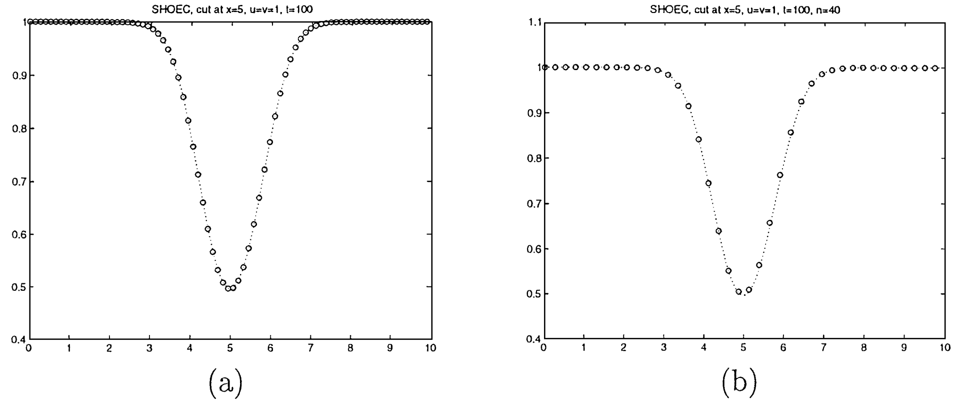
7.1. A traveling isentropic vortex. We simulate the Euler equations on a square two-dimensional domain $(x, y) \in (0, 10) \times (0, 10)$ with the initial conditions

$$\bar{u}_\infty = \left[1, u_\infty, v_\infty, \frac{1}{\gamma - 1}\right]^{\mathrm{T}},$$

perturbed by

$$[\delta u, \delta v, \delta T] = \left[-\frac{\epsilon}{2\pi} e^{(1-r^2)/2}, \frac{\epsilon}{2\pi} e^{(1-r^2)/2}, -\frac{(\gamma - 1)}{2\gamma}\left(\frac{\epsilon}{2\pi} e^{(1-r^2)/2}\right)^2\right].$$

Here, $T \equiv p/\rho$, $(\bar{x}, \bar{y}) = (x - c, y - c)$, and $r^2 = \bar{x}^2 + \bar{y}^2$. We require the flow to be isentropic, so $p\rho^{-\gamma} = \text{const} \equiv 1$.



(a) (b)

FIG. 3. Cut through density plot at $x = 5$ for $T = 100$: (a) $n = 80$; (b) $n = 40$.

The resulting state is given by

$$[\rho, \rho u, \rho v, p] \\ = \left[\left(1 - C_2\, e^{(1-r^2)}\right)^{1/(\gamma-1)}, \rho\left(u_\infty - C_1\, e^{(1-r^2)/2}\overline{y}\right), \rho\left(v_\infty + C\, e^{(1-r^2)/2}\overline{x}\right)\rho^\gamma\right].$$

This state represents a vortex that is centered at (c, c). The exact solution of the Euler equations with these initial conditions is a vortex of the same width and strength traveling with the constant speed (u_∞, v_∞). The solution is computed on a square $n \times n$ grid with periodic boundary conditions in both x- and y-directions. The constants C_1 and C_2 are chosen as $C_1 = 2.5/\pi$ and $C_2 = C_1^2/7$, and $c = 5$. We consider a diagonally traveling vortex $(u_\infty = v_\infty = 1)$. We compute the solution for the diagonally traveling vortex with the sixth-order SHOEC scheme at $t = 100$ (10 periods), and observe the decay in the quality of the solution. The results are shown in Fig. 3. The dotted lines in the graphs show the initial and exact solutions. For both $n = 80$ and $n = 40$ the solutions are very accurate, even after long time integration. We compare our results to the results obtained by Shu [37] for four different numerical methods in Fig. 4: a second-order TVD scheme (MUSCLE with the min–mod limiter [42]); a third-order ENO scheme [35]; a third-order discontinuous Galerkin method [10]; and a fifth-order weighted ENO scheme [18].

The sixth-order SHOEC scheme performs as well as the third-order discontinuous Galerkin method (DG-3) and better than the fifth-order weighted ENO scheme (WENO-5), also for $n = 40$. Clearly, the second-order TVD scheme and third-order ENO scheme result in poorer solutions. High-order accurate methods require a much lower grid density to obtain the same solution quality. In [15] we investigated the convergence of the SHOEC method using this test problem, which showed the expected order of accuracy of six.

7.2. Flow past a cylinder. The simulation of flow past a cylinder has been used extensively as a test problem for numerical schemes in two dimensions for both inviscid and viscous compressible flows. The simplicity of the geometry is deceptive; the flows are by no means simple to compute. The behavior of the inviscid flow depends on the free stream Mach number M_∞. For low Mach numbers, the flow is steady and symmetric about the horizontal and the vertical. The flow is accelerated over the cylinder and joins smoothly at the rear. As M_∞ increases, the flow eventually develops a supersonic region around the top and bottom of the cylinder. The critical Mach number M_{cr} at which this occurs first is

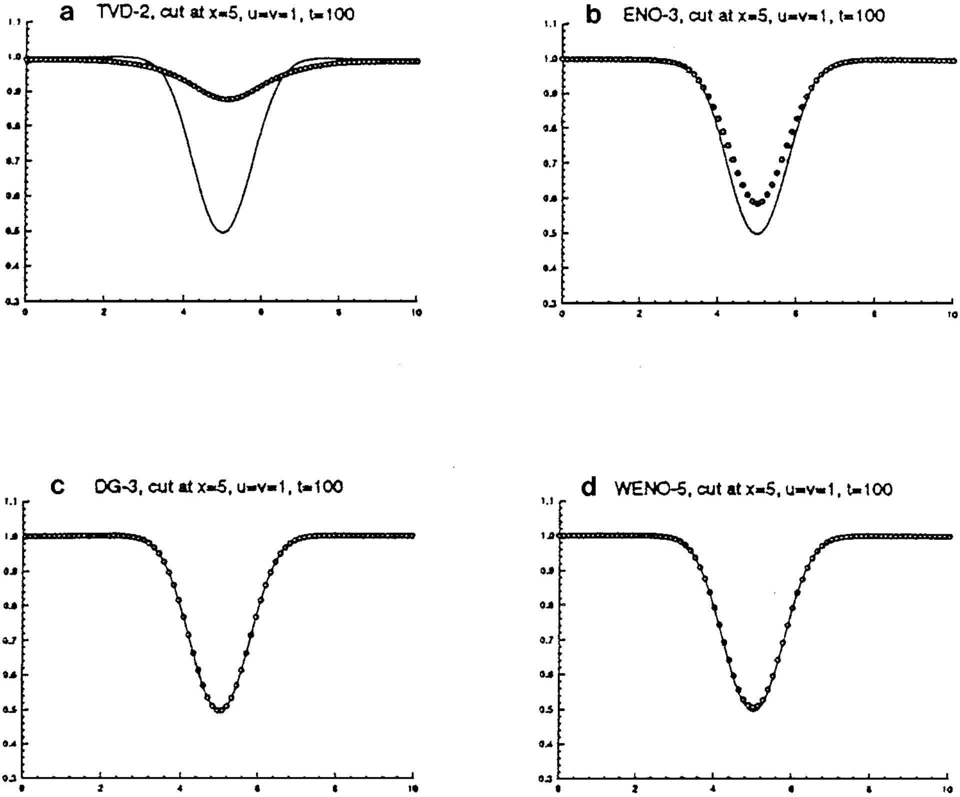


FIG. 4. Results experiments Shu for $n = 80$ and $T = 100$: (a) second-order TVD scheme; (b) third-order ENO; (c) third-order discontinuous Galerkin; (d) fifth-order WENO.

approximately $\mathrm{M}_{\mathrm{cr}} = 0.4$. Radial shock waves are formed on the lee side of the supersonic region. Because of the pressure gain over the shock, the flow separates at the lee side and circulation bubbles are formed in the wake (Fig. 5), with nearly constant pressure. The separation points are denoted by P_s.

The initial condition we use is derived from the velocity field of an incompressible potential flow past a circular cylinder with radius R, given by

$$[u, v] = \left[1 + R^2\frac{(y^2 - x^2)}{(x^2 + y^2)^2}, -R^2\frac{2xy}{(x^2 + y^2)^2}\right].$$

The expressions for the remaining variables ρ and E are found by requiring constant

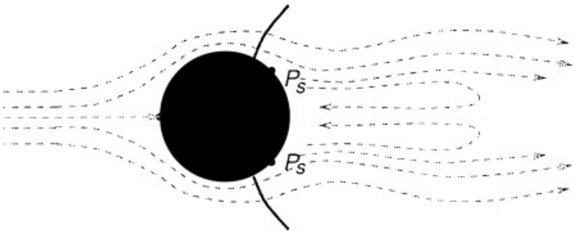


FIG. 5. Super critical ($\mathrm{M}_\infty > 0.4$) flow with stagnation points $\bar{P}_s$.

entropy in the flow domain so that $p\rho^{-\gamma} = C_1$, constant total enthalpy which requires $(E + p)/\rho = C_2$, and the uniform far field

$$\bar{u}_\infty = \left[1, 1, 0, \frac{1}{\gamma(\gamma - 1)} \frac{1}{M_\infty^2} + \frac{1}{2}\right]. \tag{7.1}$$

The far field corresponds to the velocity field for $((x^2 + y^2) \to \infty)$ with Mach number M_∞. We get

$$\rho = \left(\frac{(\gamma - 1)\left(C_2 - \frac{1}{2}(u^2 + v^2)\right)}{\gamma C_1}\right)^{1/(\gamma-1)}, \quad p = C_1\rho^\gamma.$$

We took $\mathrm{M}_\infty = 0.50$, resulting in a maximum Mach number at steady state of approximately $\mathrm{M}_{\mathrm{max}} = 1.6$, leading to the formation of two (relatively weak) shocks. We again remark that the proposed method is not designed for strong shock problems and high Mach number flows. The purpose of this experiment is to show that the proposed tools (detection, adaptation, and efficient artificial viscosity) can handle transonic flow problems well.

Because composite grids are not necessarily needed in this test we can compare composite grid results to single grid results. The composite grids used were formed by a circular component grid around the cylinder with outer radius $r = 15$ and 32×64 grid points, and a Cartesian background grid covering the square $(-40, 40) \times (-40, 40)$ with 200×200 grid points. The single circular grid has an outer radius of $r = 40$ and 32×120 grid points. The results were identical.

In Fig. 6 the evolution of the Mach contours are displayed for the first 1600 time steps. It clearly shows the formation of the lee shocks. At approximately 1700 time steps the detection algorithm reports the shock which leads to refinement in the shock regions. In this experiment, the buffer zone is equal to 4 and the whole lee side of the cylinder is regridded.

The shocks are nearly stationary; they form at 60° counting from the rear stagnation point and travel upstream as the back pressure builds up to a final position at approximately 75°. We added second-order artificial viscosity given by 5.4 in the vicinity of the shock on the fine grid. We added it in the circumferential direction only because the shock is nearly aligned with the radial grid lines. Its effectiveness is illustrated in Fig. 7. Figures 7a and c show contours close to the shock on the upper part of the cylinder at 1900 and 2300 time steps, respectively, when the SHOEC scheme is applied without artificial viscosity. Figures 7b and d show the solutions at the same time when artificial viscosity is added after time step 1700. We obtain an approximate one-point shock which is maintained at long time integration (Fig. 8). We remark that the number of contour lines is an artifact of the visualization package. At convergence the correct maximum Mach number is observed. The shock resolution compares well to the (mostly upwind) methods reported in the GAMM workshop.

Two circulation bubbles are created in the wake. The back pressure pushes the stagnation points upstream as conjectured. A close look at the circulation bubbles is provided by Fig. 9.

Pandolfi and Larocca [28] reported that in their numerical simulations for the symmetry of the flow eventually breaks down after long time integration. The same behaviour was observed in experiments run at the GAMM workshop [13] held in Rocquencourt. We did not observe this behaviour which is likely due to the high order of the SHOEC method and the small amount of artificial viscosity introduced.

In Table I we show profiling data representative of all simulations run on a Dec Alpha. The Runge–Kutta column displays the time to compute the flux derivatives in symmetrized

TABLE I
Profiling Data for a Typical SHOEC Run

	Runge–Kutta	$\tilde{f}, \tilde{g}, w$	$\tilde{f}_w, \tilde{g}_w$	Regridding	Projection	Misc.	(Overlap)
CPU%	65	7	13	2	1	11	1

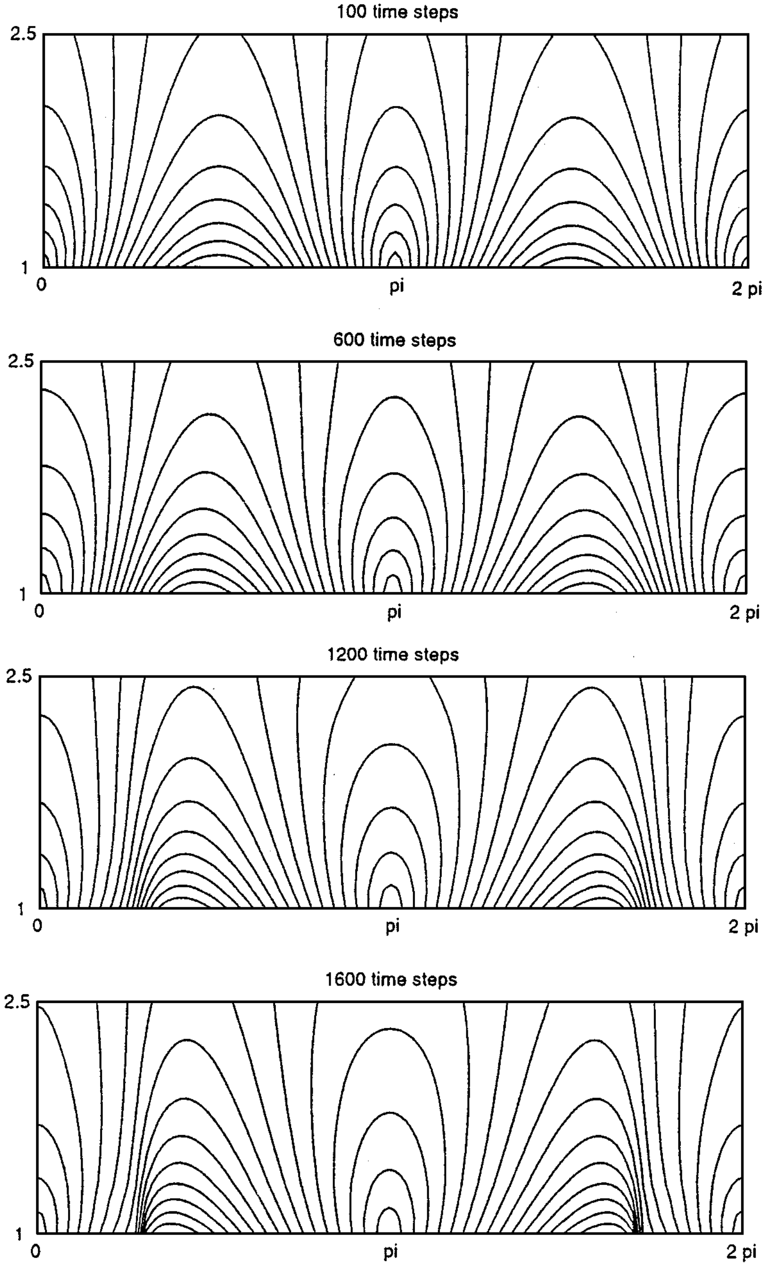


FIG. 6. Evolution Mach contours for $M_\infty = 0.50$.

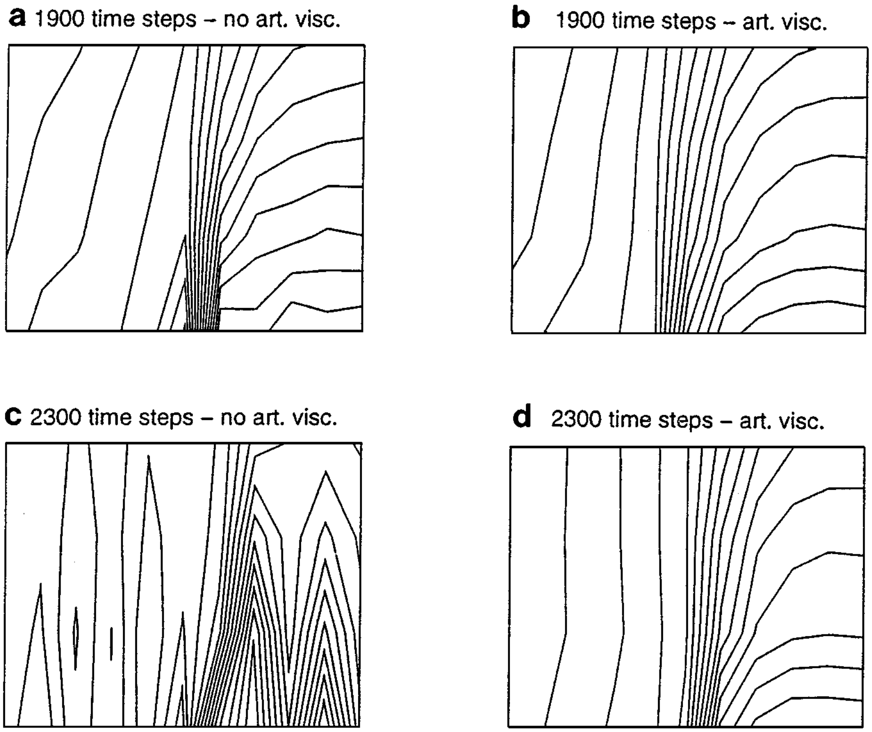


FIG. 7. Mach contours: (a) + (c), no artificial viscosity; (b) + (d), artificial viscosity. added

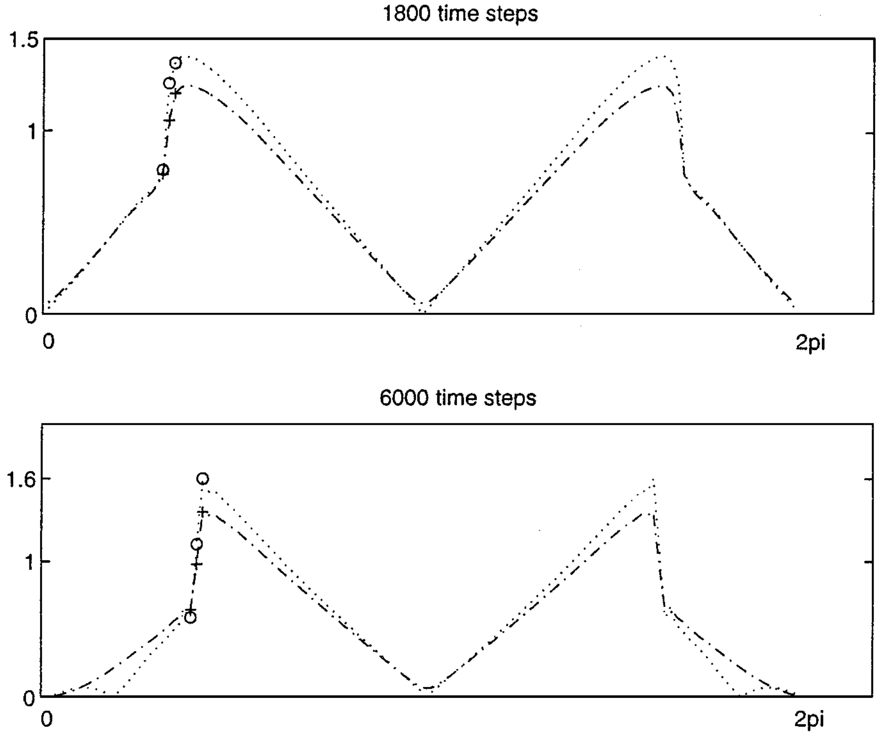


FIG. 8. Mach contour on cylinder (dotted line) and at 1 grid point distance from the cylinder (dashed line) after 1800 and 6000 time steps.

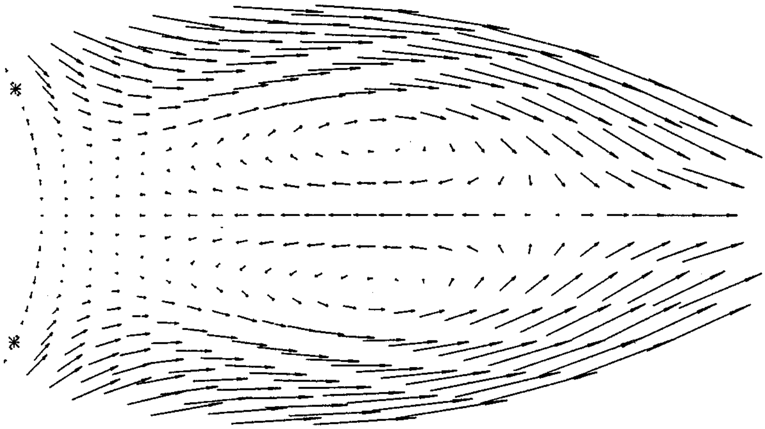


FIG. 9. Velocity field in the wake.

split form. The computations of the flux vectors $\tilde{f}$ and $\tilde{g}$, entropy variables w, and Jacobians $\tilde{f}_w$ and $\tilde{g}_w$ are timed separately. The regridding procedure, which includes detection, error control, and construction and initialization of the finer grids, takes a mere 2%. The overlap overhead due to the interpolation is around 1% of the total CPU time. We remark that in our experiments local grid adaptation reduced the computational time by a factor of 2 on average. Also, a factor of 2.5 was gained in computational efficiency by increasing the order of accuracy of the method from 2 to 4 on uniform grids. These studies were performed on subsonic flow with a source term which allowed us to derive an exact solution and accurately estimate errors.

8. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper we proposed a simple and efficient solution strategy for solving fluid flows governed by the compressible Euler equations. The general solution framework is formed by a stable and central finite difference method of high order of accuracy and implemented on composite adaptive grids. This combination is effective for solving smooth flows on complicated domains. To handle weak to moderate shocks two new tools were constructed: a fast and accurate wavelet-based detection algorithm and an effective and cheap artificial viscosity. Local grid adaptation was implemented to increase resolution where required, while keeping the overall grid density low. The high order of accuracy of the SHOEC scheme combined with the local grid adaptation approach leads to increased efficiency because coarser grids can be used in smooth regions. A particular example in the Euler test suite showed a factor of six improvement in computational time when using a fourth-order method on a two-dimensional adaptive grid, compared to a second-order method on a uniform grid. Just adapting the grid locally gave a factor of 2 improvement for the high-order method. These gains will be larger in three-dimensional applications.

The overhead of the composite adaptive grid method which includes the overlap interpolation, the detection, and the construction and initialization of the finer grids, was negligible compared to the computational kernel.

The grids were refined in the shock regions to retain accuracy because the accuracy at discontinuities is at most first order, independently of the numerical method used. Spurious oscillations generated by the high-order central schemes were suppressed with a cost-effective scalar artificial viscosity. The artificial viscosity is only added to the scheme on the finer grid(s) in the vicinity of the shock. Artificial viscosity was not required outside shock regions because of the inherent stability of the finite difference method. The viscosity added is smaller than traditionally chosen viscosities and leads to very sharp (approximately one-point) shock resolutions in both one and two dimensions.

The detection algorithm locates shocks, regions of sharp gradients, and spurious oscillations. In two dimensions it also determines the orientation of shocks. If a shock is sufficiently aligned with the grid this information can be used to perform grid refinement only in the direction normal to the shock. The detection algorithm is fast with only $O(n)$ computations needed, where n is the number of grid points. No a priori knowledge of the shock location is needed and the resulting method is therefore shock capturing. The algorithm is independent of the numerical method and can therefore be used in other solution packages.

This paper is the first stepping stone in the realization of the proposed solution strategy. We feel that the results warrant future research. The error estimation procedure is still to be implemented into the code and is nontrivial for high-order methods. In smooth regions and for short time steps the errors are likely to behave in a nonlinear fashion similar to the solution. We could, therefore, postulate a nonlinear estimate by translating the stability estimates satisfied by the solution. Tests should be run to compare the linear method against the nonlinear approach.

The detected shock orientation could be used to restrict local grid refinement to the direction normal to the shock if the shock and the grid are sufficiently aligned. The fast composite grid generator XCOG could be incorporated into the code so that component grids can be aligned with shocks or other phenomena at runtime.

APPENDIX A: SYMMETRIZATION

For $h(S) = e^{S/(\alpha+\gamma)}$,

$$w = \frac{p^*}{p}\left[E + \frac{\alpha - 1}{\gamma - 1}p \quad -\rho u \quad -\rho v \quad \rho\right]^{\mathrm{T}}$$

and

$$u_w = \frac{1}{p^*}\begin{bmatrix} a\rho & a\rho u & a\rho v & \frac{a}{2}\rho(u^2+v^2) - \frac{1}{\gamma-1}p \\ & a\rho u^2 - p & a\rho uv & u\left(\frac{a}{2}\rho(u^2+v^2) - bp\right) \\ & & a\rho v^2 - p & v\left(\frac{a}{2}\rho(u^2+v^2) - bp\right) \\ & & & -\frac{b}{\gamma-1}\frac{p^2}{\rho} - bp(u^2+v^2) + \frac{a}{4}\rho(u^2+v^2)^2 \end{bmatrix}.$$

where

$$p^* = \chi e^{S/(\alpha+\gamma)} = \chi(p\rho^{-\gamma})^{1/(\alpha+\gamma)}, \quad p = \chi^{-\beta}\left((p^*)^{\alpha} w_3^{\gamma}\right)^{1/(1-\gamma)}, \tag{A.1}$$

with $\chi = -K/\beta$. The variable p^* satisfies an equation similar to the equation of state (2.2),

$$p^* = \frac{(\gamma - 1)}{\alpha}\left(w_1 - \frac{1}{2}\frac{w_2^2}{w_3}\right).$$

Its computation through (A.1) is cheapest for $\alpha = 1 - 2\gamma$ which is what we use in our simulations. The constants a, b, and c are $a = (1 - \alpha - \gamma)/\alpha$, $b = \gamma/(\gamma - 1)$, and $c = (1 - 2\gamma)/(\gamma - 1)$. The flux vectors are

$$\tilde{f}(w) = \frac{p}{p^*}\left[-w_2 \quad \frac{w_2^2}{w_4} + p^* \quad \frac{w_2 w_3}{w_4} \quad -\frac{w_2}{w_4}\left(w_1 + \frac{\gamma - \alpha}{\gamma - 1}p^*\right)\right]^{\mathrm{T}},$$

$$\tilde{g}(w) = \frac{p}{p^*}\left[-w_3 \quad \frac{w_2 w_3}{w_4} \quad \frac{w_3^2}{w_4} + p^* \quad -\frac{w_3}{w_4}\left(w_1 + \frac{\gamma - \alpha}{\gamma - 1}p^*\right)\right]^{\mathrm{T}},$$

The upper triangular part of the symmetric matrix $\tilde{f}(w)_w$ expressed in the variables u is given by

$$\tilde{f}_w = \frac{1}{p^*} \times \begin{bmatrix} a\rho u & a\rho u^2 - p & a\rho uv & u\left(\frac{a}{2}\rho(u^2+v^2) - bp\right) \\ & u(a\rho u^2 - 3p) & v(a\rho u^2 - p) & -b\frac{p^2}{\rho} + cpu^2 + \frac{a}{2}\rho(u^2+v^2)u^2 - \frac{1}{2}p(u^2+v^2) \\ & & u(a\rho v^2 - p) & uv\left(cp + \frac{a}{2}\rho(u^2+v^2)\right) \\ & & & u\left(bc\frac{p^2}{r} + cp(u^2+v^2) + \frac{a}{4}\rho(u^2+v^2)^2\right) \end{bmatrix},$$

and for $\tilde{g}(w)_w$ by

$$\tilde{g}_w = \frac{1}{p^*} \times \begin{bmatrix} a\rho v & a\rho uv & a\rho v^2 - p & v\left(\frac{a}{2}\rho(u^2+v^2) - bp\right) \\ & v(a\rho u^2 - p) & u(a\rho v^2 - p) & uv\left(cp + \frac{a}{2}\rho(u^2+v^2)\right) \\ & & v(a\rho v^2 - 3p) & -b\frac{p^2}{\rho} + cpv^2 - \frac{1}{2}p(u^2+v^2) + \frac{a}{2}\rho v^2(u^2+v^2) \\ & & & v\left(bc\frac{p^2}{r} + cp(u^2+v^2) + \frac{a}{4}\rho(u^2+v^2)^2\right) \end{bmatrix}.$$

REFERENCES

1. M. Aftosmis, J. Melton, and M. Berger, *Adaptation and Surface Modeling for Cartesian Mesh Methods*, AIAA Paper 95-1725, 1995.
2. M. Berger, *Adaptive Mesh Refinement for Hyperbolic Differential Equations*, Ph.D. thesis, Computer Science Department, Stanford University, 1982.
3. M. Berger, On conservation at grid interfaces, *SIAM J. Numer. Anal.* **24** (1987).
4. M. Berger and J. Melton, An accuracy test of a cartesian grid method for steady flow in complex geometries, in *Proc. 5th Int. Conf. Hyperbolic Problems, 1994*.
5. M. J. Berger, Stability of interfaces with mesh refinement, *Math. Comput.* **45**, 301 (1985).
6. J. G. Blom and J. G. Verwer, A vectorizable adaptive grid solver for pdes in 3D. Part I. Algorithmic aspects and applications, *Appl. Numer. Math.* **16**, 129 (1994).
7. J. G. Blom, R. A. Trompert, and J. G. Verwer, *VLUGR2: A vectorizable adaptive grid solver for PDEs in 2D*, Technical Report NM-R9403, Department of Numerical Mathematics, Center for Mathematics and Computer Science, 1994.
8. D. Burgess, *Parallel Computing for Unstructured Mesh Algorithms.* Ph.D. thesis, Oxford University, 1996.

9. M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S Don, The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: A study of the boundary error, *SIAM J. Sci. Comput.* **16**, No. 6 (1995).

10. B. Cockburn and C.-W. Shu, The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework, *Math. Comput.* **52** (1989).

11. P. Colella and P. Woodward, The piece-wise parabolic method (ppm) for gasdynamical simulations, *J. Comput. Phys.* **54** (1984).

12. I. Daubechies, *Ten Lectures on Wavelets* (SIAM, 1992).

13. A. Dervieux, B. van Leer, J. Periaux, and A. Rizzi (Eds.), *GAMM Workshop on Numerical Simulation of Compressible Euler Flows, INRIA, 1986* (GAMM, Vieweg, 1986).

14. R. J. DiPerna, Convergence of approximate solutions to conservation laws, *Arch. Rat. Mech. Anal.* **82**, 70 (1983).

15. M. G. Gerritsen, *Designing an efficient solution strategy for fluid flows*, Ph.D. thesis, SCCM Program, Stanford University, November 1996.

16. M. G. Gerritsen and P. Olsson, Designing an efficient strategy for fluid flows. I. A high-order stable finite difference scheme and sharp shock resolution, *J. Comput. Phys.* November (1996).

17. A. Harten, S. Osher, B. Engquist, and S. Chakravarthy, Some results on uniformly high-order accurate essentially non-oscillatory schemes, *Appl. Numer. Math.* **2** (1986).

18. G. S. Jiang and C.-W. Shu, Efficient implementation of weighted eno schemes, *J. Comput. Phys.* **126**(1), 202 (1996).

19. H.-O. Kreiss and G. Scherer, On the existence of energy estimates for difference approximations for hyperbolic systems, Technical report, Department of Scientific Computing, Uppsala University, 1977.

20. R. LeVeque, *Numerical Methods for Conservation Laws*, 2nd ed. (Birkhaeuser, Basel, 1992).

21. D. Lindström, *Accurate Numerical Solution of Hyperbolic PDEs with Source Terms*, Ph.D. thesis, Department of Scientific Computing, Uppsala University, June 1996.

22. S. Mallat, Multiresolution approximation and wavelets. *Trans. Amer. Math. Soc.* **315**, 69 (1989).

23. S. Mallat and W. L. Hwang, Singularity detection and processing with wavelets, *IEEE Trans. Inform. Theory* **38**, 617 (1992).

24. S. Mallat and S. Zhong, Characterization of signals from multiscale edges, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(7), 47 (1992).

25. P. Olsson, *Energy and Maximum Norm Estimates for Nonlinear Conservation Laws*, Technical Report RIACS 94-01, Research Institute of Advanced Computer Science, 1994.

26. P. Olsson, Summation by parts, projections and stability, *Math. Comput.* **64**, 1035 (1995).

27. P. Olsson, Summation by parts, projections and stability II, *Math. Comput.* **64**, 1473 (1995).

28. M. Pandolfi and F. Larocca, Transonic flow about a circular cylinder, *Comput. & Fluids* **17** (1989).

29. E. Pärt-Enander and B. Sjögreen, Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems, *Comput. & Fluids* **23** (1994).

30. Eva Pärt-Enander, *Overlapping Grids and Applications in Gas Dynamics*, Ph.D. thesis, Uppsala University, Department of Scientific Computing, 1996.

31. N. A. Petersson, XCOG: A new algorithm for generating overlapping grids, Technical report, CHALMERS, Sweden, 1996.

32. N. A. Petersson and J. F. Malmliden, The 3-D overlapping grid generator CHALMESH, Technical report, Prosolvia Consulting, Gothenburg, Sweden, 1996.

33. Pj. Ponenti and J. Liandrat, *Numerical Algorithms Based on Biorthogonal Wavelets*, Technical Report ICASE 96-13, NASA Langley Research Center, 1996.

34. M. D. Salas and A. Iollo, *Entropy Jump across an Inviscid Shock Wave.* Technical Report ICASE 95-12, NASA Langley Research Center, 1995.

35. C. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, *J. Comput. Phys.* **77** (1988).

36. C.-W. Shu, Total-variation-diminishing time discretizations, *SIAM J. Sci. Statist. Comput.* **9**(6), 1073 (1988).

37. C.-W. Shu, private communication, September 1996.

38. B. Sjögreen, private communication, 1996.

39. G. Starius, Composite mesh difference methods for elliptic boundary value problems, *Numer. Math.* **28**, 243 (1977).

40. G. Starius, On composite mesh difference methods for hyperbolic differential equations, *Numer. Math.* **35**, 241 (1980).

41. B. Strand, Summation by parts for finite difference approximations for d/dx, *J. Comput. Phys.* **110**(1), 47 (1994).

42. B. van Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method, *J. Comput. Phys.* **32** (1979).

43. J. Walden, *Wavelet Solvers for Hyperbolic PDEs*, Ph.D. thesis, Department of Scientific Computing, Uppsala University, 1996.

44. X. Zhu, *Composite Adaptive Grid Methods for Partial Differential Equations*, Ph.D. thesis, Scientific Computing and Computational Mathematics, Stanford University, 1996.