AIAA Journal
Vol. 32, No. 2, February 1994

Efficient Cell-Vertex Upwind Scheme for the Two-Dimensional Euler Equations

C.-C. Rossow*
Deutsche Forschungsanstalt für Luft- und Raumfahrt, Braunschweig D-38108, Germany

The accuracy and efficiency of a cell-vertex upwind scheme are investigated. Upwinding is achieved by an unequal distribution of the transformed flux balances to the vertices of a computational cell. The equations are rotated with respect to the local velocity vector to determine the upwinding directions. Fourth differences background dissipation is added to ensure convergence, and time integration is performed using a five-stage Runge-Kutta scheme. The background dissipation is scaled by the appropriate eigenvalues, and convergence toward steady state is accelerated by employing upwind implicit residual smoothing and multigrid. The scheme is second order accurate in space at steady state, and it yields very accurate results on significantly coarse meshes. Thus its efficiency is competitive with that of central-differencing schemes using multigrid acceleration.

Introduction

DURING the last years, considerable effort has been concentrated on the development of cell-vertex upwind schemes. The basic idea of all schemes developed thus far is to employ an unequal distribution of the flux balances to the vertices of the computational cells. Differences arise from the particular strategy employed to achieve the distribution of flux balances to the vertices. In Ref. 1 a multidimensional approach according to Hirsch et al.[2] was used to achieve a diagonalization of the flux Jacobians independently of the direction of the grid coordinates. Simple wave solutions to the governing equations were considered in Ref. 3 to obtain the direction of wave propagation to be used for the distribution process. A splitting in streamwise and crossflow directions was employed in Refs. 4 and 5 to determine the weighting coefficients. These methods have been called fluctuation splitting,[3] upwind control volumes,[4] or flux balance splitting[5] schemes. Concerning the use of the flow direction as guidance for the upwinding process, it is interesting to note that streamwise and crossflow directions were already used in cell-centered schemes to establish grid-independent upwind schemes.[6,7]

For those interested in steady-state solutions, an essential feature of numerical schemes is the convergence toward steady state. This becomes crucial for upwind schemes, since upwind operators usually require more computational work per grid point. Slow convergence can make the use of such schemes prohibitive for practical use, even if they provide more accurate solutions than central-differencing methods. This holds especially if central-differencing schemes are combined with a multigrid acceleration technique, since the rapid convergence established with these methods allows the use of fine meshes to counterbalance the lower accuracy. Therefore, care must be taken to establish adequate acceleration techniques to enhance the convergence of upwind schemes, if they should be competitive with central-differencing schemes.

In this contribution, the efficiency of the flux balance splitting cell-vertex upwind scheme according to Ref. 5 will be enhanced. The fourth differences background dissipation needed in the scheme will be replaced by a matrix dissipation to establish scaling of the dissipation by the appropriate eigenvalues. The convergence of the scheme will be augmented by the implementation of a recently developed implicit upwind residual smoothing technique following Ref. 8. Increasing the convergence properties is vital since the computational work involved in the upwinding process of the flux balances is considerable. Furthermore, the multigrid technique is used to accelerate convergence. The combination of matrix dissipation, upwind implicit smoothing, and multigrid will be investigated, and a comparison with a central-differencing cell-vertex scheme with multigrid acceleration following Ref. 9 will be made.

Basic Idea of Flux Balance Splitting

The basic idea of flux balance splitting will be explained by an outline of the one-dimensional flux balance splitting scheme according to Ref. 5. For the one-dimensional case, no conceptual difficulties arise, since the one-dimensional Euler equations may be completely diagonalized. The one-dimensional Euler equations in conservation form read

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{1}$$

and the flux Jacobian $\bar{\bar{A}}$ is defined by

$$\frac{\partial F}{\partial x} = \bar{\bar{A}} \frac{\partial W}{\partial x}, \qquad \bar{\bar{A}} = \frac{\partial F}{\partial W}$$

Using the left and right modal matrices $\bar{\bar{M}}^{-1}$ and $\bar{\bar{M}}$, which consist of the left and right eigenvectors of the flux Jacobian $\bar{\bar{A}}$, the system of Eq. (1) may be decoupled using the transformation

$$\bar{\bar{M}}^{-1} \frac{\partial W}{\partial t} + \bar{\bar{\Lambda}} \bar{\bar{M}}^{-1} \frac{\partial W}{\partial x} = 0 \tag{2}$$

where $\bar{\bar{\Lambda}}$ is the diagonal matrix containing the eigenvalues of $\bar{\bar{A}}$. Since $\bar{\bar{\Lambda}} = \bar{\bar{M}}^{-1}\bar{\bar{A}}\bar{\bar{M}}$

$$\bar{\bar{M}}^{-1} \frac{\partial W}{\partial t} + \bar{\bar{M}}^{-1} \frac{\partial F}{\partial x} = 0 \tag{3}$$

In a one-dimensional cell-vertex scheme, the discrete flow quantities are located at the nodes of the one-dimensional grid, as sketched in Fig. 1. The discretization of Eq. (1) for each cell $i + ½$ with volume $V_{i+½}$ yields

$$\left(\frac{\mathrm{d}}{\mathrm{d}t} W\right)_{i+½} + \frac{1}{V_{i+½}} Q_{i+½} = 0 \tag{4}$$

The conservative flux balance $Q_{i+½}$ of cell $i + ½$ is defined by

$$Q_{i+½} = F_{i+1}S_{i+1} - F_iS_i \tag{5}$$

Received Nov. 1, 1992; presented as Paper 93-0071 at the AIAA 31st Aerospace Sciences Meeting, Reno, NV, Jan. 11-14, 1993; revision received April 15, 1993; accepted for publication June 23, 1993.

*Research Scientist, Institut für Entwurfsaerodynamik. Member AIAA.

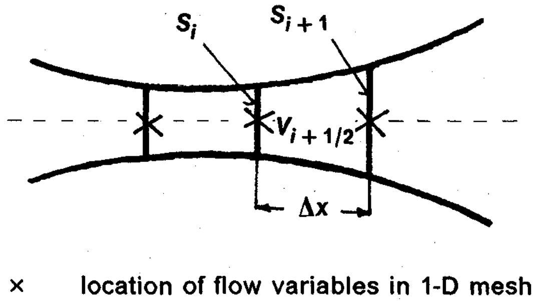

Fig. 1 Sketch of one-dimensional control volume.

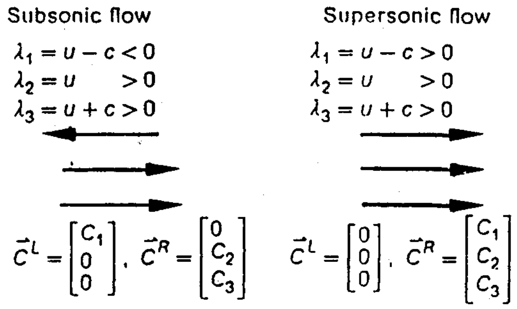
Subsonic flow
$\lambda_1 = u - c < 0$
$\lambda_2 = u > 0$
$\lambda_3 = u + c > 0$
$\vec{C}^L = \begin{bmatrix} C_1 \\ 0 \\ 0 \end{bmatrix}, \vec{C}^R = \begin{bmatrix} 0 \\ C_2 \\ C_3 \end{bmatrix}$
Supersonic flow
$\lambda_1 = u - c > 0$
$\lambda_2 = u > 0$
$\lambda_3 = u + c > 0$
$\vec{C}^L = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \vec{C}^R = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}$

Fig. 2 Distribution of characteristic flux balances.

where S_i is the area of the cross section at node i. The conservative flux balances $Q_{i+½}$ are transformed into characteristic flux balances using the local left modal matrix $\bar{\bar{M}}^{-1}$ according to Eq. (3):

$$C_{i+½} = \bar{\bar{M}}^{-1}_{i+½} \cdot Q_{i+½} \tag{6}$$

Each cell sends its characteristic flux balances $C_{i+½}$ to the downwind node. The downwind node is determined by the sign of the eigenvalue corresponding to the particular characteristic flux balance. Figure 2 gives a sketch of distributions of characteristic flux balances for subsonic and supersonic flows. The distribution step provides weighted characteristic flux balances $C^L_{i+½}$ and $C^R_{i+½}$ for the left and right node, respectively. From these, weighted conservative flux balances may be calculated by the inverse transformation with the right modal matrix $\bar{\bar{M}}$:

$$\begin{aligned} Q^L_{i+½} &= \bar{\bar{M}}_{i+½} \cdot C^L_{i+½} \\ Q^R_{i+½} &= \bar{\bar{M}}_{i+½} \cdot C^R_{i+½} \end{aligned} \tag{7}$$

The contributions of left and right cells are summed up to yield the conservative flux balance Q_i at a node, and thus the numerical analog of Eq. (1) at a node i reads

$$\left(\frac{\mathrm{d}}{\mathrm{d}t} W\right)_i + \frac{1}{V_i}\left(Q^R_{i-½} + Q^L_{i+½}\right) = 0 \tag{8}$$

where $V_i = 0.5\ (V_{i-½} + V_{i+½})$.

The scheme is formally first order accurate; however, in Ref. 5, it was shown that at steady state it yields second-order-accurate solutions, despite the use of one-sided, two-point-difference formulas. It usually captures a shock within one cell without any pre- and postshock oscillations.

Solution of the Two-Dimensional Euler Equations

Governing Equations

The two-dimensional Euler equations in conservation form are given by

$$\frac{\partial W}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{9}$$

where the flux Jacobians are defined by

$$\frac{\partial F}{\partial x} = \bar{\bar{A}}_x \frac{\partial W}{\partial x}, \qquad \bar{\bar{A}}_x = \frac{\partial F}{\partial W}$$

$$\frac{\partial G}{\partial y} = \bar{\bar{A}}_y \frac{\partial W}{\partial y}, \qquad \bar{\bar{A}}_y = \frac{\partial G}{\partial W}$$

In contrast to the one-dimensional Euler equations, a complete decoupling of the system of Eqs. (9) is not possible, since the matrices $\bar{\bar{A}}_x$ and $\bar{\bar{A}}_y$ do not commute. Physically, this may be explained by observing that in multidimensional flows information is propagated in infinitely many directions. Nevertheless, Eq. (9) may be transformed with respect to arbitrary directions. For example, a transformation with respect to the x direction using the left modal matrix $\bar{\bar{M}}_x^{-1}$ of the flux Jacobian $\bar{\bar{A}}_x$ yields

$$\bar{\bar{M}}_x^{-1} \frac{\partial W}{\partial t} + \bar{\bar{M}}_x^{-1}\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y}\right) = 0 \tag{10}$$

Spatial Discretization

For the two-dimensional scheme a structured mesh with quadrilateral cells is considered. The flow quantities are located at the cell vertices, as sketched in Fig. 3. Discretization of Eq. (9) for a cell $i + ½, j + ½$ yields

$$\left(\frac{\mathrm{d}}{\mathrm{d}t} W\right)_{i+½,j+½} + \frac{1}{V_{i+½,j+½}} Q_{i+½,j+½} = 0 \tag{11}$$

where the numerical integration to obtain the flux balance $Q_{i+½,\ j+½}$ is evaluated by the trapezoidal rule. Since the two-dimensional Euler equations may not be completely decoupled, no mathematical guidance for the choice of appropriate directions for the distribution of the flux balances toward the cell vertices is available. According to Ref. 5 the local velocity vector is chosen to determine the upwinding directions. The equations are rotated into a coordinate frame aligned with the local flow velocity. The flux balances $Q_{i+½,j+½}$ are transformed into characteristic flux balances using the left modal matrix $\bar{\bar{M}}_s^{-1}$ of the streamwise direction. The characteristic flux balances are distributed to the vertices of the particular cell using the sign of the corresponding eigenvalues. After the distribution step, the inverse transformation is performed with the streamwise right modal matrix $\bar{\bar{M}}_s$. but only contributions to the continuity, streamwise momentum, and energy equations are retained; the contribution to the crossflow momentum equation is discarded. The whole process is repeated using the crossflow direction, but after the inverse transformation all contributions except those to the normal momentum equation are discarded. The scheme described thus far is very similar to the upwind control volume approach[4] for unstructured meshes. However, there no distinction is made between the upwinding in streamwise and crossflow directions, whereas in the flux balance splitting[5] the crossflow upwinding contributes only to normal momentum. Numerical experiments on the structured meshes employed here showed this strategy to be more robust. Note that for the crossflow direction the linear eigenvalues vanish, and only + c and − c remain as eigenvalues, where c is the speed of sound.

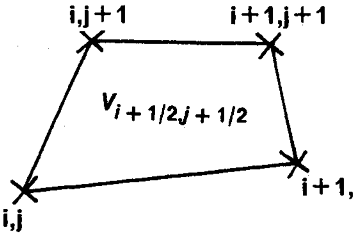

Fig. 3 Sketch of two-dimensional control volume.

To determine the weighting coefficients for the distribution of the characteristic flux balances, the coordinates of the cell vertices of a particular cell are projected onto the local velocity vector, yielding projected coordinates ξ as sketched in Fig. 4. For information propagation into streamwise direction, i.e., positive eigenvalues, the most upstream node has the weight zero, and advancing downstream the weight increases linearly. For conservation the sum of the weighting coefficients is always unity. For negative eigenvalues the weighting process is reversed. In crossflow direction the same process is applied using the direction normal to the local flow velocity.

The discrete analog of Eq. (9) at node i,j is obtained by collecting the contributions of all cells surrounding that node:

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}W\right)_{i,j} + \frac{1}{V_{i,j}}Q_{i,j} = 0 \tag{12}$$

where

$$V_{i,j} = \tfrac{1}{4}(V_{i-\frac{1}{2},j-\frac{1}{2}} + V_{i+\frac{1}{2},j-\frac{1}{2}} + V_{i+\frac{1}{2},j+\frac{1}{2}} + V_{i-\frac{1}{2},j+\frac{1}{2}})$$

and $Q_{i,j}$ is given by the sum of the contributions of the four cells surrounding node i,j.

In the one-dimensional flux balance splitting scheme, the discrete equations have been directly integrated in time. However, in the two-dimensional case, numerical oscillations occurred that prevented convergence to machine zero or even lead to divergence. Since the linear eigenvalues in the crossflow direction vanish, the damping provided by the remaining two equations with eigenvalues $+c$ and $-c$ is insufficient. Furthermore, in contrast to the one-dimensional scheme, positivity is not insured (see Ref. 3), and the integration of Eq. (12) may become unstable. To remedy this problem, an artificially constructed dissipative term has to be added, and in Ref. 5 fourth differences of the dependent variables are used to provide a third-order background dissipation. Denoting the dissipative term by $\boldsymbol{D}$, Eq. (12) then becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{W}_{i,j} = -\frac{1}{V_{i,j}}(\boldsymbol{Q}_{i,j} - \boldsymbol{D}_{i,j}) = -\boldsymbol{R}_{i,j} \tag{13}$$

with $\boldsymbol{R}_{i,j}$ called the residual at node i,j. The dissipative term reads

$$\boldsymbol{D}_{i,j} = (-\boldsymbol{D}_\xi^4 - \boldsymbol{D}_\eta^4)\boldsymbol{W}_{i,j} \tag{14}$$

where ξ and η are the curvilinear coordinates in the direction of the computational coordinates i and j, respectively. The fourth difference operator in the ξ direction, for example, is defined as

$$\boldsymbol{D}_\xi^4\boldsymbol{W}_{i,j} = \nabla_\xi\left\{[\lambda_{i+\frac{1}{2},j}\epsilon^{(4)}_{i+\frac{1}{2},j}]\Delta_\xi\nabla_\xi\Delta_\xi\right\}\boldsymbol{W}_{i,j} \tag{15}$$

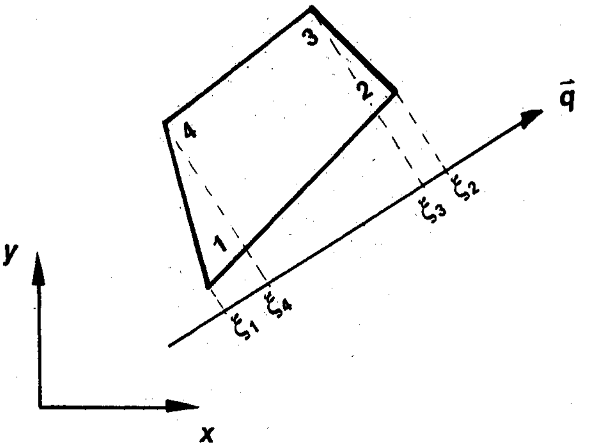

Fig. 4 Projection of coordinates on flow direction.

where Δ_ξ and ∇_ξ are forward and backward difference operators in the ξ direction. The operator for the η direction is defined in a similar manner. In the original scheme[5] the scaling factor λ is chosen according to the work of Jameson et al.[10] as

$$\lambda_{i+\frac{1}{2},j} = [(\lambda_\xi)_{i+\frac{1}{2},j} + (\lambda_\eta)_{i+\frac{1}{2},j}] \tag{16}$$

where λ_ξ and λ_η are proportional to the largest eigenvalues of the Jacobians $\bar{A}_\xi$ and $\bar{A}_\eta$:

$$\begin{aligned}(\lambda_\xi)_{i+\frac{1}{2}j} &= |\boldsymbol{q}_{i+\frac{1}{2}j}\cdot\boldsymbol{S}_{\xi;\,i+\frac{1}{2},j}| + c|\boldsymbol{S}_{\xi;\,i+\frac{1}{2}j}|\\(\lambda_\eta)_{i+\frac{1}{2}j} &= |\boldsymbol{q}_{i+\frac{1}{2}j}\cdot\boldsymbol{S}_{\eta;\,i+\frac{1}{2},j}| + c|\boldsymbol{S}_{\eta;\,i+\frac{1}{2}j}|\end{aligned} \tag{17}$$

where $\boldsymbol{q}$ is the local velocity vector, and $\boldsymbol{S}_\xi$ and $\boldsymbol{S}_\eta$ represent cell face vectors. The coefficient $\epsilon^{(4)}$ in Eq. (15) contains a user-specified constant $k^{(4)}$ to control the amount of dissipation, and via a pressure switch the fourth differences dissipation is switched off at shocks. Note that in contrast to the usual dissipation model of Ref. 10 no second differences dissipation is included, since the distribution of flux balances accounts for the shock capturing.

In this study the scalar dissipation used in Ref. 5 is replaced by a matrix dissipation model according to Ref. 11. The scalar $\lambda_{i+\frac{1}{2},j}$ containing the largest eigenvalue is replaced by matrices corresponding either to the ξ or η direction:

$$\begin{aligned}\boldsymbol{D}_\xi^4\boldsymbol{W}_{i,j} &= \nabla_\xi\left\{[\bar{\bar{A}}_{\xi;\,i+\frac{1}{2}j}\epsilon^{(4)}_{i+\frac{1}{2},j}]\Delta_\xi\nabla_\xi\Delta_\xi\right\}\boldsymbol{W}_{i,j}\\\boldsymbol{D}_\eta^4\boldsymbol{W}_{i,j} &= \nabla_\eta\left\{[\bar{\bar{A}}_{\eta;\,i,j+\frac{1}{2}}\epsilon^{(4)}_{i,j+\frac{1}{2}}]\Delta_\eta\nabla\eta\Delta_\eta\right\}\boldsymbol{W}_{i,j}\end{aligned} \tag{18}$$

The matrices $\bar{\bar{A}}_\xi$ and $\bar{\bar{A}}_\eta$ are defined using the Jacobians $\bar{A}_\xi$ and $\bar{A}_\eta$ corresponding to the ξ and η directions, respectively:

$$\begin{aligned}\bar{\bar{A}}_{\xi;\,i+\frac{1}{2},j} &= |\bar{\bar{A}}_{\xi;\,i+\frac{1}{2},j}\cdot\boldsymbol{S}_{\xi;\,i+\frac{1}{2},j}|\\\bar{\bar{A}}_{\eta;\,i,j+\frac{1}{2}} &= |\bar{\bar{A}}_{\eta;\,i,j+\frac{1}{2}}\cdot\boldsymbol{S}_{\eta;\,i,j+\frac{1}{2}}|\end{aligned} \tag{19}$$

Employing a matrix-valued dissipation, one finds that the amount of artificial viscosity is now individually scaled by the eigenvalues corresponding to the characteristic equations of the different coordinate directions. Consequently, the amount of dissipation is considerably lower compared with the scaling with the largest eigenvalue; in combination with a central-differencing discretization its use closely resembles an upwind scheme. The use of a matrix instead of a scalar requires additional computational work; however, in Ref. 11 a very efficient way to achieve this multiplication has been derived. In Ref. 5 it was shown that even with the addition of the dissipative operator the flux balance splitting scheme is still second order accurate at steady state.

Time-Stepping Scheme

The system of ordinary differential equations that is obtained by the space discretization is advanced in time by a five-stage Runge-Kutta scheme. The dissipative operator is evaluated only in the first stage and then is frozen for the remaining stages. The steady-state solution is independent of the time step, and therefore the scheme is amenable to convergence acceleration techniques. The methods employed are local time stepping, implicit residual smoothing, and multigrid.

With local time stepping, the solution at each mesh point is advanced at the maximum Δt allowed by stability.

The stability range of the basic time-stepping scheme can be extended using implicit smoothing of residuals. In Ref. 5 the technique recommended in Ref. 12 was used. This smoothing technique replaces the residual of a given grid point by a centrally weighted average of the neighboring residuals. In the i direction, the smoothing reads

$$-\epsilon\tilde{\boldsymbol{R}}_{i-1,j} + (1+2\epsilon)\tilde{\boldsymbol{R}}_{i,j} - \epsilon\tilde{\boldsymbol{R}}_{i+1,j} = \boldsymbol{R}_{i,j} \tag{20}$$

where $\tilde{\boldsymbol{R}}$ denotes the smoothed residual and ϵ is the smoothing coefficient. In two dimensions the smoothing is applied in product form, i.e., the residual obtained from the smoothing in the i direction is smoothed with the same technique in the j direction. The use of this central smoothing technique permits roughly doubling the CFL number of the basic explicit time-stepping scheme, and CFL numbers of about 7.5 are used in combination with a five-stage Runge-Kutta scheme.

To improve both damping and convergence characteristics of multistage time-stepping schemes, particularly in connection with upwind spatial differencing operators, scientists have recently developed upwind-biased implicit residual smoothing techniques (see Ref. 8). In contrast to the centrally weighted residual smoothing, the residuals are smoothed using a one-sided operator based on wave propagation directions. This can be accomplished by splitting the residual in Eq. (20):

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} W_{i,j} + \boldsymbol{R}_{i,j}^{+} + \boldsymbol{R}_{i,j}^{-} &= 0 \\ \boldsymbol{R}_{i,j}^{+} + \boldsymbol{R}_{i,j}^{-} &= \boldsymbol{R}_{i,j} \end{aligned} \tag{21}$$

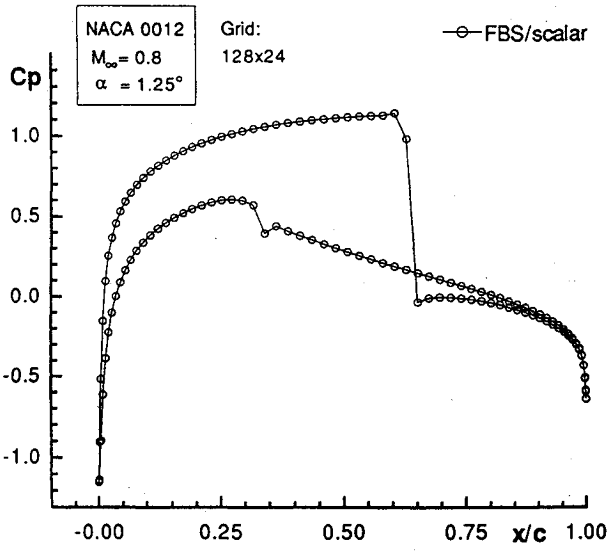

Fig. 5 Pressure distribution of FBS with scalar dissipation.

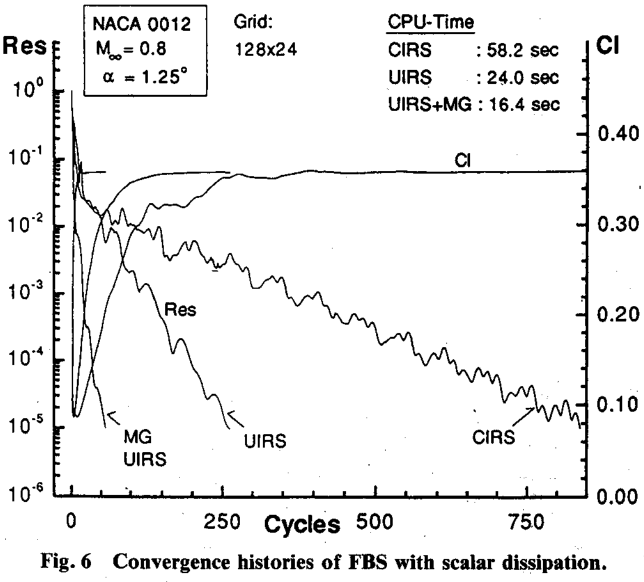

Fig. 6 Convergence histories of FBS with scalar dissipation.

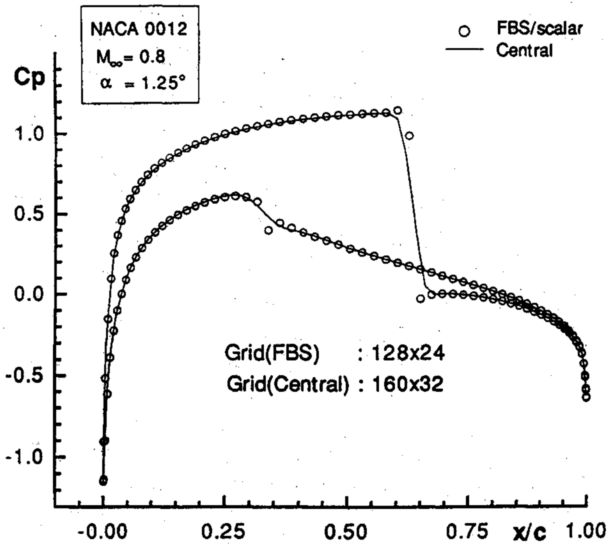

Fig. 7 Comparison of pressure distribution (FBS/central).

where $\boldsymbol{R}^+$ and $\boldsymbol{R}^-$ denote contributions of the residual $\boldsymbol{R}$ corresponding to positive and negative eigenvalues of the flux Jacobians $\bar{\bar{A}}_\xi$ or $\bar{\bar{A}}_\eta$, depending on the coordinate direction in which the smoothing is applied. For a smoothing in the i direction, the term $\boldsymbol{R}^+$ is calculated as

$$\boldsymbol{R}_{i,j}^{+} = \bar{\bar{A}}_{\xi;\, i,j}^{+} \boldsymbol{R}_{i,j} = (\bar{\bar{M}}_{\xi;\, i,j} \bar{\bar{\Lambda}}_{i,j}^{+} \bar{\bar{M}}_{\xi\,+\, i,j}^{-1})\, \boldsymbol{R}_{i,j} \tag{22}$$

where $\bar{\bar{\Lambda}}^+$ is a diagonal matrix containing the positive eigenvalues of the Jacobian $\bar{\bar{A}}_\xi$ all set to 1 and the negative eigenvalues set to zero; $\boldsymbol{R}^-$ is evaluated by $\boldsymbol{R}^- = \boldsymbol{R} - \boldsymbol{R}^+$.

Upwind-implicit smoothing is now applied independently to $\boldsymbol{R}^+$ and $\boldsymbol{R}^-$:

$$\begin{aligned} -\epsilon\tilde{\boldsymbol{R}}_{i-1,j}^{+} + (1+\epsilon)\tilde{\boldsymbol{R}}_{i,j}^{+} &= \boldsymbol{R}_{i,j}^{+} \\ (1+\epsilon)\tilde{\boldsymbol{R}}_{i,j}^{-} - \epsilon\tilde{\boldsymbol{R}}_{i+1,j}^{-} &= \boldsymbol{R}_{i,j}^{-} \end{aligned} \tag{23}$$

Equation (23) can easily be solved by backward and forward substitution, respectively. After smoothing positive and negative components of the residual, these are simply added to give the complete, smoothed residual

$$\boldsymbol{R}_{i,j} = \boldsymbol{R}_{i,j}^{+} + \boldsymbol{R}_{i,j}^{-} \tag{24}$$

The splitting of the residual in Eq. (22) can be performed efficiently using the technique given in Ref. 11. In combination with a five-stage scheme using optimized coefficients for a first-order upwind scheme according to Ref. 13 and the flux balance splitting discretization, CFL numbers of 20.5 are obtained.

For further convergence acceleration, a multigrid technique according to Ref. 14 is used. For the multigrid process, coarser meshes are created by eliminating successively every second mesh line in each coordinate direction. The solution is transferred to coarser meshes by injection. Residuals are transferred from fine to coarse meshes by a weighted average over the fine mesh points closest to the point on the coarse mesh. A forcing function is constructed such that the solution on the coarse mesh is driven by the residuals collected on the fine mesh. This procedure is repeated until the coarsest mesh is reached. Then, corrections are transferred to the next finer mesh using bilinear interpolation. A fixed W-cycle with four grid levels is used to execute the multigrid strategy. The discretization with the flux balance splitting technique combined with the upwind-implicit smoothing is applied on all four grid levels.

Results

Results will be shown for the transonic flow around the NACA 0012 airfoil at $M_\infty = 0.8$ and $\alpha = 1.25$ deg. For these freestream conditions, a shock of considerable strength occurs

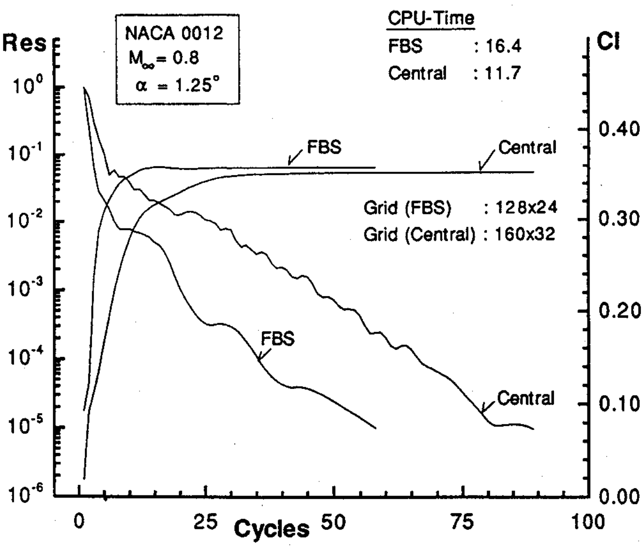

Fig. 8 Comparison of convergence histories (FBS/central).

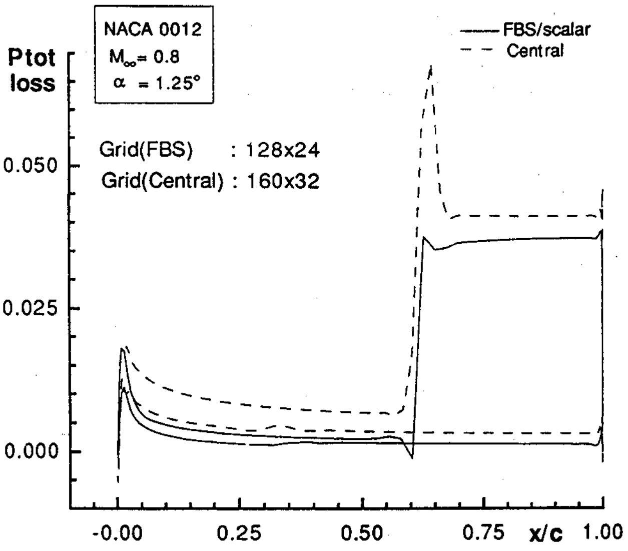

Fig. 9 Comparison of total pressure losses (FBS/central).

on the upper surface, whereas on the lower surface a very weak shock develops. This problem serves as a well-suited test case since it will reveal the properties of a particular scheme to capture shocks of different strength. Besides the ability to capture shock waves, the accuracy of a numerical scheme can be assessed by considering total pressure losses. Ahead of the shock waves, total pressure losses (or gains) should be zero, and their occurence may be attributed directly to numerical errors. In the following, first the flux balance splitting scheme with scalar background dissipation will be investigated in combination with upwind implicit smoothing and multigrid. Second, the influence of the matrix dissipation on accuracy and convergence will be shown. For comparison, results of the cell-vertex central-differencing scheme with multigrid acceleration of Ref. 9 will be used. All calculations were performed on a Cray YMP computer.

Flux Balance Splitting with Scalar Dissipation

Figure 5 displays the pressure distribution for the NACA 0012 at $M_\infty = 0.8$ and $\alpha = 1.25$ deg obtained with the flux balance splitting (FBS) scheme. The computational grid has an O-topology with 128 cells around the airfoil and 24 cells in normal direction. The user-specified constant $k^{(4)}$ of the fourth differences background dissipation was set to 1/64. The shocks on the upper surface as well as the weak shock on the lower surface are both captured within almost one cell. In Fig. 6 the convergence histories of the FBS scheme with different acceleration techniques are shown. The computation was stopped when the residual decreased to 10^{-5}. The use of upwind-implicit smoothing (UIRS) reduces the number of time steps by roughly a factor of 3 compared with the case with central-implicit smoothing (CIRS). This reduction corresponds almost directly to the increase of the CFL number from 7.5 to 20.5. Both smoothing techniques were employed in each stage of the Runge-Kutta scheme, and the smoothing coefficients were set to $\epsilon = 0.5$ and 1.5 for CIRS and UIRS, respectively.

Implementing multigrid additionally reduces the number of required cycles by almost a factor of 4 compared with UIRS and a factor of 14 compared with CIRS. In the figure a table with computation times is included. Applying UIRS reduced the necessary computation time by a factor of 2 and the multigrid by roughly a factor of 4 compared with the basic scheme.

Figure 7 shows pressure distributions obtained with FBS and with a central-difference cell-vertex scheme. The results of the central-differencing scheme are, however, obtained on a 160 × 32 mesh. The coefficients of the artificial dissipation are set to $k^{(4)} = 1/64$ for the fourth differences and to $k^{(2)} = 1$ for the second differences dissipation necessary to capture shock

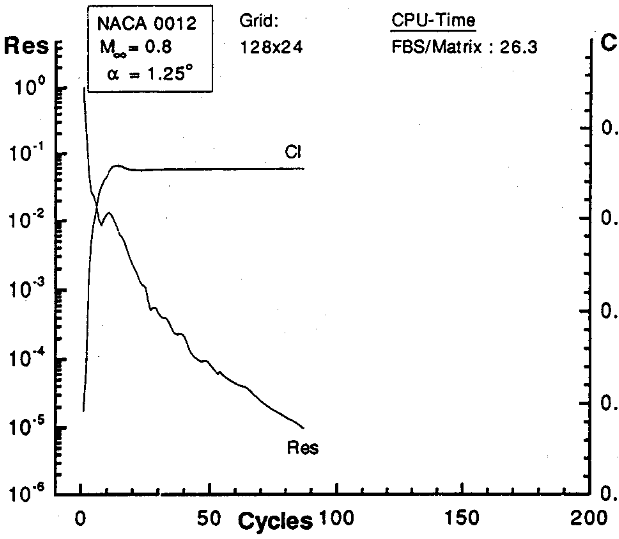

Fig. 10 Convergence history of FBS with matrix dissipation.

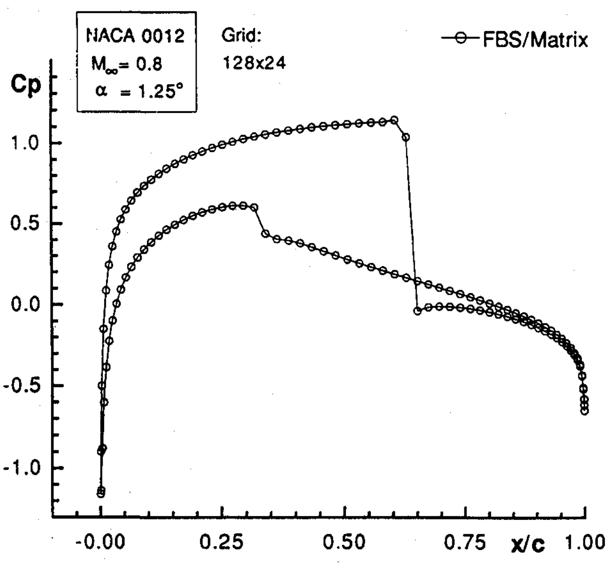

Fig. 11 Pressure distribution of FBS with matrix dissipation.

waves with the central scheme. In spite of the finer mesh, the shocks are still smeared by the central scheme, especially the weak shock on the lower surface. Figure 8 shows a comparison of the convergence rates for the methods on these meshes. Because of multigrid acceleration, convergence rates of the schemes do not differ significantly. The somewhat better convergence rate of the FBS scheme is caused by the use of upwind-implicit smoothing. A comparison of computation times shows that the FBS scheme requires about 1.5 times the CPU time needed by the central scheme. Figure 9 displays total pressure losses on the airfoil contour for both schemes. In general it can be stated that the accuracy of the FBS scheme is higher compared with the central scheme; only at the leading edge do both schemes show the same peak in total pressure rising to about 1.8%.

Flux Balance Splitting with Matrix Dissipation

When the matrix dissipation is used, the amount of numerical viscosity is reduced due to the scaling with appropriate eigenvalues, and more accurate solutions are expected. However, reducing the amount of numerical dissipation also reduces the high-frequency damping properties, which are essential for the multigrid algorithm. Figure 10 shows convergence rates for the FBS scheme and matrix dissipation with implicit upwind smoothing and multigrid. The constant $k^{(4)}$ is kept to 1/64 as in the scalar case. For the multigrid scheme, the number of required cycles increased from 58 for the scalar dissipation to 87 for the matrix dissipation. The execution of one time step with matrix dissipation required only about 10% more CPU time than in case of using scalar dissipation.

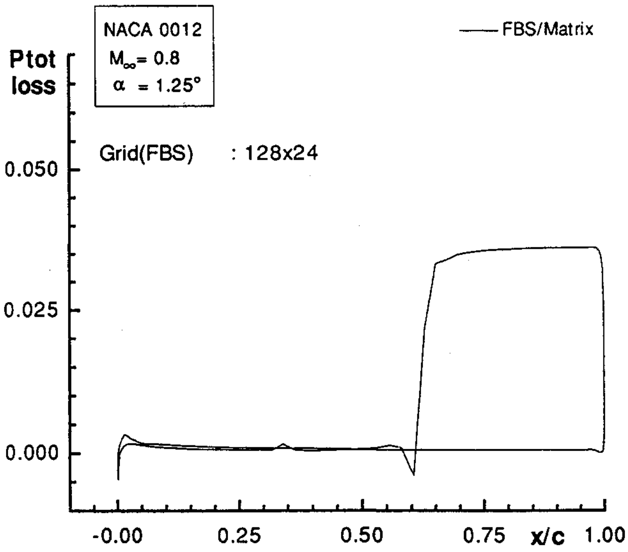

Fig. 12 Total pressure losses of FBS with matrix dissipation.

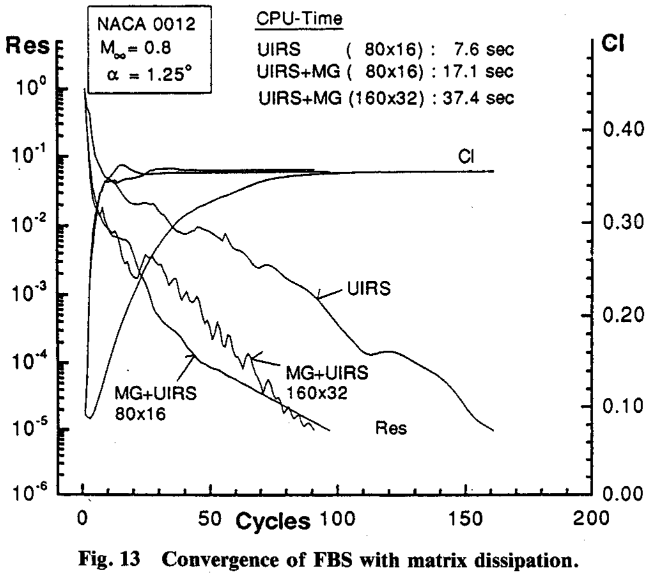

Fig. 13 Convergence of FBS with matrix dissipation.

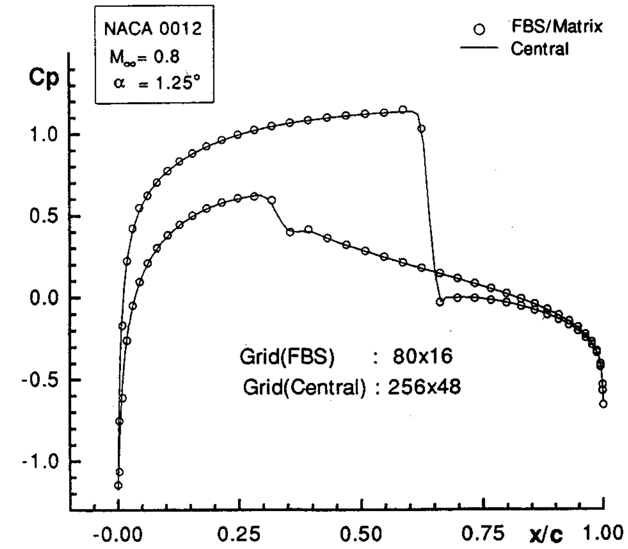

Fig. 14 Pressure distribution for FBS (coarse mesh)/central (fine mesh).

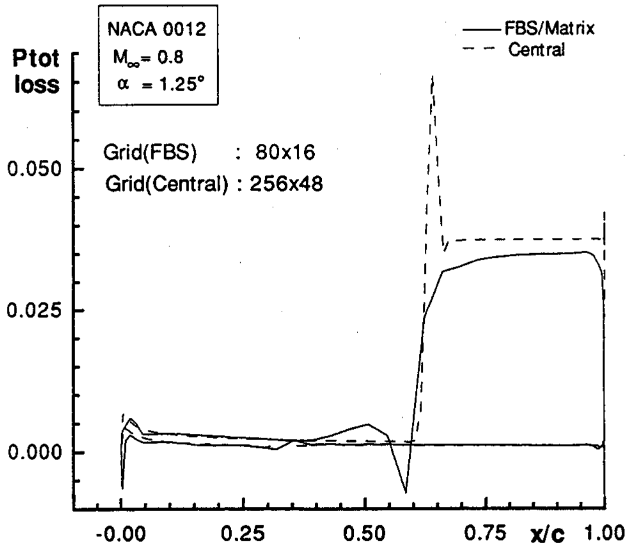

Fig. 15 Total pressure losses for FBS (coarse mesh)/central (fine mesh).

Figure 11 shows the pressure distribution obtained with matrix dissipation. Note that the shock capturing is almost not affected, since the fourth differences dissipation is switched off at shocks. Regarding the total pressure losses in Fig. 12, the effect of matrix dissipation becomes clearly visible. Compared with the scalar dissiption, total pressure losses are reduced by a factor of 3 to 4.

The results using matrix dissipation demonstrated the benefit in accuracy; however, the price is a degradation of convergence due to the reduced high-frequency damping. To exploit the higher accuracy, a mesh of 80 × 16 cells was used. Figure 13 shows the convergence rates using UIRS with and without the multigrid. The convergence with multigrid is similar to that obtained on the 128 × 24 grid. However, expressed in terms of CPU time, the multigrid offers no advantage with respect to UIRS alone. Because of this coarse mesh, large time steps are possible, and UIRS alone provides already good convergence rates. Since the CPU time needed for a W-multigrid cycle is considerable compared with a single mesh time step, on this coarse mesh UIRS alone performs better with respect to CPU time than with multigrid acceleration. However, with multigrid the convergence does not degrade when finer meshes are employed, as can be seen from the convergence history of UIRS with multigrid on a 160 × 32 mesh, which has been included in Fig. 13. Therefore, on finer meshes the multigrid still offers its advantages.

The pressure distribution obtained with FBS for this case on the 80 × 16 mesh is displayed in Fig. 14. The shocks are still captured within one cell. Also displayed are the results of the central-differencing scheme on a mesh with 256 × 48 mesh points. The results of the FBS scheme on the 80 × 16 mesh agree favorably with these results.

Figure 15 shows total pressure losses computed with both schemes on the different meshes. The numerical errors in both cases are in the range of 0.5% at the leading edge and between 0.1 and 0.2% ahead of the shock. Figures 14 and 15 show that with FBS the same accuracy may be obtained on a mesh with about 10 times less mesh points than required for the central-differencing scheme. Computation time for the FBS scheme was 7.6 s with UIRS alone compared with 25 s for the central scheme with the multigrid, where convergence was reached after 99 cycles.

Conclusions

The accuracy and efficiency of a cell-vertex upwind scheme according to Ref. 5 were investigated and improved. The convergence of the scheme was substantially enhanced by applying an implicit upwind residual smoothing technique and a multigrid algorithm. Furthermore, the original scalar fourth differences background dissipation was replaced by a matrix dissipation. The scheme had remarkable shock-capturing capabilities, even for very weak shocks. Since the background dissipation is switched off at shocks, the shock-capturing properties are almost independent from the background dissipation. The matrix dissipation reduced the total pressure losses on the contour by factors of 3 to 4. Application of the upwind-implicit smoothing allowed CFL numbers of 20.5 to be used, thus reducing the number of time steps required to reach steady state by a factor of 3. Especially in combination with the scalar dissipation, the multigrid further enhanced the efficiency, and convergence rates comparable to central schemes with multigrid were established. The use of matrix dissipation slightly degraded the convergence rates due to the lower high-frequency damping. In a final study, a very coarse mesh was used to demonstrate the benefit in accuracy provided by the flux balance splitting scheme with matrix dissipation. Even on this coarse mesh, total pressure losses due to numerical errors were less than 1%. Compared with a central-differencing cell-vertex scheme, a mesh with almost 10 times fewer grid points could be used to obtain the same level of accuracy. This leads to a reduction in CPU time by a factor of 3.

References

[1]Powell, K. G., and Van Leer, B., "A Genuinely Multi-Dimensional Upwind Cell Vertex Scheme for the Euler Equations," AIAA Paper 89-0095, Jan. 1989.

[2]Hirsch, C., Lacor, C., and Deconinck, H., "Convection Algorithms Based on a Diagonalization Procedure for the Multidimensional Euler Equations," AIAA Paper 87-1163, June 1987.

[3]Struijs, R., Deconinck, H., and Roe, P. L., "Fluctuation Splitting for the 2-D Euler Equations," VKI Lecture series 1991-01, von Kármán Inst., Brussels, Belgium, March 1991.

[4]Giles, M., Anderson, W., and Roberts, T., "Upwind Control Volumes: A New Approach," AIAA Paper 90-0104, Jan. 1990.

[5]Rossow, C.-C., "Flux Balance Splitting with Rotated Differences: A Second Order Accurate Cell Vertex Upwind Scheme," *Proceedings of the 9th GAMM Conference on Numerical Methods in Fluid Mechanics,* edited by J. B. Vos, A. Rizzi, and I. L. Ryhming, Vol. 35, Vieweg Verlag, Braunschweig, Germany, Sept. 1991, pp. 567–576.

[6]Goorjian, P. M., "Algorithm Developments for the Euler Equations with Calculations of Transonic Flows," AIAA Paper 87-0536, Jan. 1987.

[7]Levy, D. W., Powell, K. G., and Van Leer, B., "An Implementation of a Grid-Independent Upwind Scheme for the Euler Equations," AIAA Paper 89-1931, June 1989.

[8]Blazek, J., Kroll, N., Radespiel, R., and Rossow, C.-C., "Upwind Implicit Residual Smoothing Method for Multi-Stage Schemes," AIAA Paper 1533, June 1991.

[9]Radespiel, R., Rossow, C.-C., and Swanson, R. C., "Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier-Stokes Equations," *AIAA Journal,* Vol. 28, No. 8, 1990, pp. 1464–1472.

[10]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes," AIAA 81-1259, June 1981.

[11]Turkel, E., "Improving the Accuracy of Central Difference Schemes," *Proceedings of the 11th Conference on Numerical Methods in Fluid Dynamics,* Vol. 323, Lecture Notes in Physics, Springer-Verlag, Berlin, 1988, pp. 586–591.

[12]Jameson, A., and Baker, T. J., "Solution of the Euler Equations for Complex Configurations," AIAA Paper 83-1929, June 1983.

[13]Van Leer, B., Tai, C.-H., and Powell, K. G., "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations," AIAA Paper 89-1933, June 1989.

[14]Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," Princeton Univ., Mechanical and Aerospace Engineering Rept. 1743, Princeton, NJ; also 2nd European Conf. on Multigrid Methods, Cologne, Germany, Oct. 1985.