Multigrid acceleration of a flux-difference splitting method for steady Euler equations *

Erik DICK
Department of Machinery, State University of Ghent, Sint Pietersnieuwstraat 41, B-9000 Gent, Belgium

Received 30 May 1988
Revised 17 October 1988

Abstract: A flux-difference splitting based on the polynomial character of the flux-vectors is introduced for steady Euler equations. This splitting is applied to finite volumes centered around the vertices of the computational grid. A discrete set of equations is obtained which is both conservative and positive. The flux-difference splitting is done in an algebraically exact way. As a consequence, shocks are represented as sharp discontinuities, without wiggles. Due to the positivity, the set of equations can be solved by collective relaxation methods.

A full multigrid method based on symmetric successive relaxation, full weighting, bilinear interpolation and W-cycle is presented. Typical full multigrid efficiency is achieved for the GAMM transonic bump test case since after the starting cycle and one multigrid cycle, the solution cannot be distinguished anymore from the fully converged solution.

Keywords: Flux-difference splitting, steady Euler equations, multigrid methods.

1. Introduction

The flux-vector splitting approach was introduced by Steger and Warming [9] for the unsteady Euler equations. Their splitting is based on the homogeneity of degree one with respect to the conservative variables $\rho, \rho u, \rho v, \rho E$. It was shown by Jespersen [4] that this flux-vector splitting can also be applied directly to the steady Euler equations to generate discrete equations that can be solved by relaxation methods in multigrid form. This technique, however, shows some shortcomings in the treatment of shocks. In the conservative formulation, so-called undifferenced terms appear. These terms represent a loss of positivity of the discrete set of equations and cause oscillations in the vicinity of shocks.

Going back to the earlier work of Godunov [2], a remedy for the shock oscillations can be found in not splitting the flux-vectors themselves, but differences of flux-vectors. Several flux-difference splitting procedures were proposed for unsteady equations, simplifying the Godunov method. The splitting of Roe [8] is based on the quadratic character of the flux-vectors with respect to the variables $\sqrt{\rho}$, $\sqrt{\rho} u$, $\sqrt{\rho} v$, $\sqrt{\rho} H$. The splitting of Osher [6] is based on characteristic paths and results in a splitting with respect to the variables $\sqrt{\gamma p/\rho}$, $u$, $v$, $\ln(\rho/\rho')$.

* The research reported in this paper was granted by the Belgian National Science Foundation (N.F.W.O.).
A very simple splitting based on the polynomial character of the flux-vectors with respect to the primitive variables \( \rho, u, v, p \) was proposed by Lombard et al. [5].

It was shown by Hemker and Spekreijse [3] that the Osher scheme can be applied directly to the steady Euler equations, to form a multigrid method. In this paper, a similar approach is used, but based on the simpler flux-difference splitting of Lombard et al. In contrast to their original approach, which used an approximate splitting, the splitting is done here in an algebraically exact way. This is necessary to treat the steady equations directly, avoiding the time marching necessary in the original approach. Also, due to the algebraically exact manipulation, boundary conditions can be introduced in a rigorous way.

A detailed description of the splitting technique was given by the author in [1]. In this paper, the principles of the method are summarized and the multigrid formulation is discussed.

2. Flux-difference splitting with respect to primitive variables

Steady Euler equations, in two dimensions, take the form

\[
\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0, \tag{1}
\]

where the flux-vectors are

\[
f^T = \{ \rho u, \rho uu + p, \rho uv, \rho Hu \}, \quad g^T = \{ \rho v, \rho uv, \rho vv + p, \rho Hv \}, \tag{2}
\]

\( \rho \) is density, \( u \) and \( v \) are Cartesian velocity components, \( p \) is pressure, \( H = \gamma p/(\gamma - 1)p + \frac{1}{2}u^2 + \frac{1}{2}v^2 \) is total enthalpy and \( \gamma \) is the adiabatic constant.

Since the components of the flux-vectors form polynomials with respect to the primitive variables \( \rho, u, v \) and \( p \), components of flux-differences can be written as follows:

\[
\Delta \rho u = \bar{u} \Delta \rho + \bar{p} \Delta u,
\]

\[
\Delta (\rho uu + p) = \rho \bar{u} \Delta u + \bar{u} \Delta \rho u + \Delta p = \bar{u}^2 \Delta \rho + (\bar{\rho} \bar{u} + \bar{p}) \Delta u + \Delta p,
\]

\[
\Delta \rho Hu = \bar{\rho} \bar{u} \left( \frac{1}{2} \Delta u^2 + \frac{1}{2} \Delta v^2 \right) + \frac{1}{2} (u^2 + v^2) \Delta \rho u + \frac{\gamma}{\gamma - 1} \Delta p u
\]

\[
+ \bar{\rho} \bar{u} \bar{v} \Delta v + \frac{\gamma}{\gamma - 1} \bar{u} \Delta p,
\]

where the bar denotes mean value.

With the definition of \( \bar{q}^2 = \frac{1}{2}(u^2 + v^2) \), the flux-difference \( \Delta f \) can be written as

\[
\Delta f = \begin{pmatrix}
\bar{u} & \bar{\rho} & 0 & 0 \\
\bar{u}^2 & \bar{\rho} \bar{u} + \bar{\rho} u & 0 & 1 \\
\bar{u} \bar{v} & \bar{\rho} \bar{v} & \bar{\rho} u & 0 \\
\bar{q}^2 u & \bar{q}^2 \bar{\rho} + \bar{\rho} u \bar{u} + \frac{\gamma}{\gamma - 1} \bar{p} & \bar{\rho} u \bar{v} & \frac{\gamma}{\gamma - 1} \bar{u}
\end{pmatrix} \Delta \xi.
\]

where \( \xi^T = \{ \rho, u, v, p \} \).
With the definition of \( \tilde{u} \) by \( \tilde{\rho} \tilde{u} = \rho u \), the flux-difference \( \Delta f \) is given by

\[
\Delta f = \begin{pmatrix}
1 & 0 & 0 & 0 \\
\tilde{u} & \tilde{\rho} & 0 & 0 \\
\tilde{v} & 0 & \tilde{\rho} & 0 \\
q^2 \tilde{\rho} \tilde{u} & \tilde{\rho} \tilde{v} & 1/\gamma - 1 & 0
\end{pmatrix}
\begin{pmatrix}
\tilde{u} \\
\tilde{\rho} \\
0 \\
0
\end{pmatrix}
= \Delta \xi.
\]  

(3)

By denoting the first matrix in (3) by \( T \), it is easily seen that the flux-difference \( \Delta g \) can be written in a similar way as

\[
\Delta g = T
\begin{pmatrix}
\tilde{v} & 0 & \tilde{\rho} & 0 \\
0 & \tilde{\rho} & 0 & 0 \\
0 & 0 & \tilde{\rho} & 0 \\
0 & 0 & \gamma \tilde{\rho} & \tilde{v}
\end{pmatrix}
\begin{pmatrix}
\tilde{u} \\
\tilde{\rho} \\
0 \\
0
\end{pmatrix}
= \Delta \xi.
\]  

(4)

where \( \tilde{\rho} \tilde{v} = \rho v \).

Any linear combination of \( \Delta f \) and \( \Delta g \) can be written as

\[
\Delta \phi = \alpha_1 \Delta f + \alpha_2 \Delta g = \tilde{A} \Delta \xi = T\tilde{\Lambda} \Delta \xi,
\]  

(5)

where

\[
\tilde{A} = \begin{pmatrix}
\tilde{w} & \alpha_1 \tilde{\rho} & \alpha_2 \tilde{\rho} & 0 \\
0 & \tilde{w} & 0 & \alpha_1 / \tilde{\rho} \\
0 & 0 & \tilde{w} & \alpha_2 / \tilde{\rho} \\
0 & \alpha_1 \gamma \tilde{\rho} & \alpha_2 \gamma \tilde{\rho} & \tilde{w}
\end{pmatrix},
\]  

(6)

with \( \tilde{w} - \alpha_1 \tilde{u} + \alpha_2 \tilde{v}, \tilde{w} = \alpha_1 \tilde{u} + \alpha_2 \tilde{v} \).

It is easy to verify that the matrix \( \tilde{A} \) has real eigenvalues and a complete set of eigenvectors [1]. For \( \alpha_1^2 + \alpha_2^2 = 1 \), the eigenvalues are given by

\[
\lambda_1 = \tilde{w}, \quad \lambda_2 = \tilde{w}, \quad \lambda_3 = \tilde{w} + \tilde{c}, \quad \lambda_4 = \tilde{w} - \tilde{c},
\]

where

\[
\tilde{w} = \frac{1}{2}(\tilde{w} + \tilde{w}) \quad \text{and} \quad \tilde{c}^2 = \gamma \tilde{\rho} / \tilde{\rho} + \frac{1}{4}(\tilde{w} - \tilde{w})^2.
\]

Following the procedure of Steger and Warming [9], the matrix \( \tilde{A} \) can be split into positive and negative parts by

\[
\tilde{A}^+ = X^{-1} \Lambda^+ X, \quad \tilde{A}^- = X^{-1} \Lambda^- X,
\]  

(7)

where \( X \) denotes the left eigenvector matrix and where

\[
\Lambda^+ = \text{diag}(\lambda_1^+, \lambda_2^+, \lambda_3^+, \lambda_4^+), \quad \Lambda^- = \text{diag}(\lambda_1^-, \lambda_2^-, \lambda_3^-, \lambda_4^-),
\]

with \( \lambda_i^+ = \max(\lambda_i, 0), \lambda_i^- = \min(\lambda_i, 0) \).

With positive and negative matrices, matrices with respectively nonnegative and nonpositive eigenvalues are meant.

This allows a splitting of the flux-difference (5) by:

\[
\Delta \phi = \tilde{A}^+ \Delta \xi + \tilde{A}^- \Delta \xi.
\]  

(8)
3. Construction of a positive discretization

Figure 1 shows a control volume centered around the node \((i, j)\). Also the nodes located inside the adjacent volumes are indicated.

When a piecewise constant interpolation of variables is chosen, the flux-difference over the surface \(S_{i+\frac{1}{2}}\) of the control volume can be written as

\[
\Delta F_{i,i+1} = \Delta y_{i+\frac{1}{2}} \Delta f_{i,i+1} + \Delta x_{i+\frac{1}{2}} \Delta g_{i,i+1} = \Delta x_{i+\frac{1}{2}} \left( \alpha_1 \Delta f_{i,i+1} + \alpha_2 \Delta g_{i,i+1} \right),
\]

where \(\Delta s_{i+\frac{1}{2}}^2 = \Delta x_{i+\frac{1}{2}}^2 + \Delta y_{i+\frac{1}{2}}^2\), \(\alpha_1 = \Delta y_{i+\frac{1}{2}} / \Delta s_{i+\frac{1}{2}}\), \(\alpha_2 = \Delta x_{i+\frac{1}{2}} / \Delta s_{i+\frac{1}{2}}\). With the notation of the previous section, this is:

\[
\Delta F_{i,i+1} = F_{i+1} - F_i = \Delta s_{i+\frac{1}{2}} A_{i,i+1} \Delta \xi_{i,i+1}.
\]

Furthermore, the matrix \(A_{i,i+1}\) can be split into positive and negative parts. This allows the definition of the absolute value of the flux-difference by

\[
|\Delta F_{i,i+1}| = \Delta s_{i+\frac{1}{2}} (A_{i,i+1}^+ - A_{i,i+1}^-) \Delta \xi_{i,i+1}.
\]

Based on (10) an upwind definition of the flux is:

\[
F_{i+\frac{1}{2}} = \left[ F_i + F_{i+1} - |\Delta F_{i,i+1}| \right].
\]

That this represents an upwind flux can be verified by writing (11) in either of the two following ways, which are completely equivalent:

\[
F_{i+\frac{1}{2}} = F_i + \frac{1}{2} \Delta F_{i,i+1} - \frac{1}{2} |\Delta F_{i,i+1}| = F_i + \Delta s_{i+\frac{1}{2}} A_{i,i+1}^- \Delta \xi_{i,i+1},
\]

\[
F_{i+\frac{1}{2}} = F_{i+1} - \frac{1}{2} \Delta F_{i,i+1} - \frac{1}{2} |\Delta F_{i,i+1}| = F_{i+1} - \Delta s_{i+\frac{1}{2}} A_{i,i+1}^+ \Delta \xi_{i,i+1}.
\]

Indeed, when \(A_{i,i+1}\) has only positive eigenvalues, the flux \(F_{i+\frac{1}{2}}\) is taken to be \(F_i\) and when \(A_{i,i+1}\) has only negative eigenvalues, the flux \(F_{i+\frac{1}{2}}\) is taken to be \(F_{i+1}\).

The fluxes on the other surfaces of the control volume \(S_{i-\frac{1}{2}}, S_{j+\frac{1}{2}}, S_{j-\frac{1}{2}}\), can be treated in a similar way as the flux on the surface \(S_{i+\frac{1}{2}}\). With (12) and (13), the flux-balance on the control volume.

---

Fig. 1. Control volume centered around node \((i, j)\).
volume of Fig. 1 can be brought into the following form:

\[ \Delta s_{i,j} \mathbf{A}^+_{i,j}[\xi_{j+\frac{1}{2}} - \xi_{j}] + \Delta s_{i,j}^{-\frac{1}{2}} \mathbf{A}^-_{i,j}[\xi_{i+\frac{1}{2}} - \xi_{i}] + \Delta s_{j+\frac{1}{2}} \mathbf{A}^-_{j,i}[\xi_{j+\frac{1}{2}} - \xi_{j}] + \Delta s_{j+\frac{1}{2}}^{-\frac{1}{2}} \mathbf{A}^+_{j,i}[\xi_{j+\frac{1}{2}} - \xi_{j}] = 0. \]  

The set formed by the equations (14) for all nodes is both conservative and positive. It is conservative since it exactly expresses the sums of fluxes on the control volume to be zero. It is positive since (14) can be put into the form

\[ C\xi_{i,j} = \Delta s_{i,j}^{-\frac{1}{2}} \mathbf{A}^-_{i,j}[\xi_{i-1,j} - \xi_{i,j}] + \Delta s_{i,j}^{-\frac{1}{2}} \mathbf{A}^+_{i,j}[\xi_{i+1,j} - \xi_{i,j}] + \Delta s_{j,i}^{-\frac{1}{2}} \mathbf{A}^-_{j,i}[\xi_{j,i-1} - \xi_{j,i}] + \Delta s_{j,i}^{-\frac{1}{2}} \mathbf{A}^+_{j,i}[\xi_{j,i+1} - \xi_{j,i}], \]

where \( C \) is the sum of the matrix-coefficients in the right-hand side and where these coefficients have nonnegative eigenvalues.

As a consequence of the positivity, a solution can be obtained by a collective variant of any scalar relaxation method. By a collective variant it is meant that in each node, all components of the vector of dependent variables \( \xi \) are relaxed simultaneously.

4. Boundary conditions

Figure 2 shows the half-volumes centered around a node at inlet and around a node on a solid boundary. These half-volumes can be seen as the limit of complete volumes in which one of the sides tends to the boundary. As a consequence, the flux on the side \( S_j \) of the control volume can be expressed according to (13) by

\[ F_j = \Delta s_j \mathbf{A}^+_{i,j}[\xi_j - \xi_{j-1}], \]

where the matrix \( \mathbf{A}_{i,j} \) is calculated in the node \((i, j)\).

Similarly, the flux on the side \( S_j \) of the control volume at the solid boundary can be expressed by

\[ F_j = \Delta s_j \mathbf{A}^+_{i,j}[\xi_j - \xi_{j-1}], \]

where again the matrix \( \mathbf{A}_{i,j} \) is calculated in the node \((i, j)\).
With the definitions (16) and (17), the flux-balance on the control volumes at boundaries takes the form (14) in which a node outside the domain comes in. These nodes, however, can be eliminated.

It is easily seen that on a solid boundary, three combinations of (17) exist, eliminating the outside node. These equations are to be supplemented by the boundary condition of impermeability. Similarly, at a subsonic inflow, one combination exists, eliminating the outside node. At a subsonic outflow, three combinations are found. Physically, as inlet boundary conditions, stagnation pressure, stagnation temperature and flow direction are to be prescribed. At outlet, Mach number can be prescribed. Due to the linearity of the condition of impermeability, the set of equations on a solid boundary is a quasi-linear set which is very similar to the set in the flow field. At inflow and outflow boundaries, the physical boundary conditions are highly nonlinear combinations of the primitive variables. Therefore, the introduction of the boundary conditions in the way as described above, necessitates iteration. This complicates the algorithm. Moreover, this treatment of boundary conditions is highly reflective. This has a detrimental effect on the performance of the method. Therefore, it is better to treat the nodes at inlet and outlet as auxiliary points and to determine the variables in these points by extrapolation. At inlet, Mach number is extrapolated along streamlines. Together with the given boundary conditions, this determines all flow variables in a direct way. At outflow the stagnation values and the flow direction are extrapolated along streamlines. Together with the prescribed Mach number, again this determines all flow variables in a direct way.

5. Numerical example

Figure 3 shows the well-known GAMM-test case [7] for transonic flows, discretized by a grid with $24 \times 8$ elements. In the actual computation a twice more refined grid was used with $96 \times 32$ elements. Vertex centered finite volumes, as indicated in Fig. 2, were used. At inflow, the specification of a horizontal flow direction was used as boundary condition. At outflow, the Mach number was fixed at 0.85.

Figure 4 shows the iso-Mach lines for the fully converged solution plotted by piecewise linear interpolation within the elements of the grid. Figure 5 shows the surface pressure distribution on the southern boundary. The obtained solution coincides almost with the solutions obtained from the most reliable time-marching methods reported in [7]. However, unlike most time-marching solutions, due to the guaranteed positivity everywhere, the solution has no wiggles in the shock region.

Fig. 3. GAMM-test case with $24 \times 8$-grid.
Figure 6 shows the cycle-structure of the multigrid method. Both the starting cycle and the repeated cycle have W-form. A full approximation scheme (FAS) on the nonlinear equations (15) is used. The relaxation algorithm is Gauss–Seidel. The order of relaxation is the lexicographic order, i.e., going from the lower left point to the upper right point first varying the row index and then going from the upper right point to the lower left point in the reverse order. In relaxing the set of equations (15), the coefficients are formed with the latest available information. This means that in the first sweep $A_{i,j}^{t-1}$ is evaluated with the function values in node $(i, j)$ on the old level, but with the function values in node $(i-1, j)$ on the new level. After determination of the new values in node $(i, j)$, no updates of coefficients and no extra iterations are done. This means that the set of equations (15) is treated as a quasi-linear set. As restriction operator for residuals, full weighting is used within the flow field while injection is used at the boundaries. The prolongation operator is bilinear interpolation. The restriction for function values is injection. The calculation starts from a uniform flow with Mach number 0.85 on the coarsest grid ($12 \times 4$).

In Fig. 6, the operation count is indicated. A relaxation on the current grid is taken as one local work unit. So, the symmetric relaxation is seen as two work units. A residual evaluation plus the associated grid transfer is also taken as one local work unit. Hence, the “4” in Fig. 6, in going down, stands for the construction of the right-hand side in the FAS-formulation, two
relaxations and one residual evaluation. With this way of evaluating the work, the cost of the repeated cycle is 8.6875 work units on the finest level. The cost of the starting cycle is about 4.39 work units.

Figure 7 shows the convergence behaviour of the single grid and the multigrid formulation. The residual shown is the maximum residual of all equations, after normalizing these equations, i.e., setting the coefficient of \( \rho, u, v \) and \( p \) on 1 in the mass-, momentum-x-, momentum-y- and energy-equation, respectively, and dividing the variables by their value in the initial uniform flow.

A maximum residual of \( 10^{-4} \) is reached after approximately 55 work units. The convergence factor of the multigrid method, i.e., the residual reduction per work unit is about 0.865. This probably can be considered as being optimal. This is seen by the pressure distribution on the bottom obtained after the nested iteration and one cycle. Up to plotting accuracy this pressure distribution coincides with the distribution obtained after full convergence, as shown in Fig. 5.

6. Conclusion

It has been shown that by an adequate use of the flux-difference splitting technique a simple and efficient multigrid method can be obtained for steady Euler equations. The formulation described here is only first-order accurate. A next step is of course to bring it into second-order accuracy. Also, the relaxation method used is sequential and cannot be vectorized. Preliminary experiments with vectorizable algorithms like Jacobi-relaxation and red-black relaxation show some loss in efficiency in terms of work units, but a gain with a factor larger than 10 in computing time on the 2-pipe CYBER-205.

References