INT. J. NUM. METHODS HEAT FLUID FLOW, VOL. 1, 51–62 (1991)

# MULTIGRID SOLUTION OF STEADY EULER EQUATIONS BASED ON POLYNOMIAL FLUX-DIFFERENCE SPLITTING

ERIK DICK

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

## ABSTRACT

A flux-difference splitting based on the polynomial character of the flux vectors is applied to steady Euler equations, discretized with a vertex-centred finite volume method. In first order accurate form, a discrete set of equations is obtained which is both conservative and positive. Due to the positivity, the set of equations can be solved by collective relaxation methods in multigrid form. A full multigrid method based on successive relaxation, full weighting, bilinear interpolation and W-cycle is used. Second order accuracy is obtained by the Chakravarthy–Osher flux-extrapolation technique, using the Roe–Chakravarthy minmod limiter. In second order form, direct relaxation of the discrete equations is no longer possible due to the loss of positivity. A defect-correction is used in order to solve the second order system.

KEY WORDS Steady Euler equations Flux-difference splitting Multigrid methods

## INTRODUCTION

A flux-difference splitting method based on the polynomial character of the flux-vectors was introduced by Dick[1]. This splitting is a pure Roe-type splitting, i.e. it satisfies the so-called U-properties introduced by Roe[2]. In contrast to that method, which 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 relies only on the polynomial character with respect to the primitive variables $\rho$, $u$, $v$, $p$, avoiding in this way square root evaluations. The polynomial splitting was inspired by earlier work by Lombard *et al.*[3], where the same idea was used in an approximate way. An algebraically exact formulation, however, is necessary to arrive at discrete equations which can be solved by relaxation methods. The polynomial splitting is simpler than that in Reference 2. This is mainly due to the omission of the secondary requirement that the discrete value of the variables coming into the definition of the splitting has to be unique. This condition of uniqueness defines the so-called Roe averages. In the polynomial splitting, three different average values of the convective velocity are used. The secondary requirements of uniqueness of average values is unnecessary and is demonstrated in this paper.

Flux-difference splitting is more complicated than the related flux-vector splitting but has the advantage that direct relaxation of the discrete steady equations is possible, while flux-vector splitting techniques necessitate time stepping. This is due to the so-called positive character of the discretization obtained by flux-difference splitting and non-positive character of the discretization obtained when using flux-vector splitting. Multigrid methods based on relaxation are much more efficient than multigrid methods based on time stepping due to better smoothing properties. The flux-difference splitting generally used in relaxation type multigrid methods is that of Osher–Chakravarthy[4]. Examples of the use of this technique are given by Hemker, Spekreijse and Koren[5–8] and is preferred because of its differentiability which allows local Newton linearization. It is generally believed that the quadratic convergence properties associated to

0961–5539/91/010051–12$2.00 *Received December 1990*

Newton linearization are important in reaching good multigrid efficiency. Roe-type splitting is not differentiable and as a consequence, only quasi-linearization with linear convergence behaviour is possible. In this paper, it is demonstrated that differentiability is not necessary to achieve good multigrid performance. A multigrid procedure, similar to the one used by Hemker, Spekreijse and Koren[5–8], is employed but replacing the complicated differentiable Osher–Chakravarthy flux-difference splitting by the simpler, non-differentiable polynomial flux-difference splitting. The resulting method's performance is better when measured in terms of work units, where the work unit is the computational time for one basic relaxation on the finest grid and the work unit is much cheaper.

Preliminary examples of the method used here were given earlier[9,10]. Since these publications the method has undergone further simplifications. In this paper, the principles of the method are briefly discussed and several examples showing its efficiency and the accuracy are given.

## POLYNOMIAL FLUX-DIFFERENCE SPLITTING

Steady state Euler equations, in two dimensions, take the form:

$$\frac{\partial f}{x}+\frac{\partial g}{\partial y}=0 \tag{1}$$

where the flux-vectors are:

$$f^T=\{\rho u,\rho uu+p,\rho uv,\rho Hu\},\qquad g^T=\{\rho v,\rho uv,\rho vv+p,\rho Hv\}$$

and where $\rho$ is density, $u$ and $v$ are Cartesian velocity components, $p$ is pressure, $H=\gamma p/(\gamma-1)\rho+\frac{1}{2}u^2+\frac{1}{2}v^2$ is the 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:

$$\Delta\rho u=\bar{u}\,\Delta\rho+\bar{\rho}\,\Delta u$$

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

$$\Delta\rho Hu=\overline{\rho u}\,(\tfrac{1}{2}\Delta u^2+\tfrac{1}{2}\Delta v^2)+\tfrac{1}{2}(\overline{u^2}+\overline{v^2})\Delta\rho u+\frac{\gamma}{\gamma-1}\Delta pu$$

$$=\tfrac{1}{2}(\overline{u^2}+\overline{v^2})\bar{u}\,\Delta\rho+\tfrac{1}{2}(\overline{u^2}+\overline{v^2})\bar{\rho}\,\Delta u+\overline{\rho u\,u}\,\Delta u+$$

$$\frac{\gamma}{\gamma-1}\bar{p}\,\Delta u+\overline{\rho u\,v}\,\Delta v+\frac{\gamma}{\gamma-1}\bar{u}\,\Delta p$$

etc., where the overbar denotes a mean value.

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

$$\Delta f=\begin{pmatrix}\bar{u} & \bar{\rho} & 0 & 0\\ \bar{u}^2 & \overline{\rho\,u}+\overline{\rho u} & 0 & 1\\ \overline{u\,v} & \overline{\rho\,v} & \overline{\rho u} & 0\\ \overline{q\,u} & \overline{q\,\rho}+\overline{\rho u\,u}+\dfrac{\gamma}{\gamma-1}\bar{p} & \overline{\rho u\,v} & \dfrac{\gamma}{\gamma-1}\bar{u}\end{pmatrix}\Delta\xi$$

where $\xi$ is the vector of primitive variables

$$\xi^T=\{\rho,u,v,p\}$$

Defining $\tilde{u}$ by $\bar{\rho}\tilde{u}=\overline{\rho u}$, the flux-difference $\Delta f$ is given by:

$$\Delta f=\begin{pmatrix}1 & 0 & 0 & 0\\ \tilde{u} & \bar{\rho} & 0 & 0\\ \tilde{v} & 0 & \bar{\rho} & 0\\ \bar{q} & \overline{\rho u} & \overline{\rho v} & 1/\gamma-1\end{pmatrix}\begin{pmatrix}\tilde{u} & \bar{\rho} & 0 & 0\\ 0 & \tilde{u} & 0 & 1/\bar{\rho}\\ 0 & 0 & \tilde{u} & 0\\ 0 & \gamma\bar{p} & 0 & \tilde{u}\end{pmatrix}\Delta\xi \tag{2}$$

By denoting the first matrix in (2) by $T$, it is easily seen that the flux-difference $\Delta g$ can be written in a similar way as:

$$\Delta g=T\begin{pmatrix}\tilde{v} & 0 & \bar{\rho} & 0\\ 0 & \tilde{v} & 0 & 0\\ 0 & 0 & \tilde{v} & 1/\bar{\rho}\\ 0 & 0 & \gamma\bar{p} & \tilde{v}\end{pmatrix}\Delta\xi$$

where $\bar{\rho}\tilde{v}=\overline{\rho v}$.

Any linear combination of $\Delta f$ and $\Delta g$ can be written as:

$$\Delta\phi=n_x\Delta f+n_y\Delta g=\bar{A}\Delta\xi=T\tilde{A}\Delta\xi \tag{3}$$

where

$$\tilde{A}=\begin{pmatrix}\bar{w} & n_x\bar{\rho} & n_y\bar{\rho} & 0\\ 0 & \bar{w} & 0 & n_x/\bar{\rho}\\ 0 & 0 & \bar{w} & n_y/\bar{\rho}\\ 0 & n_x\gamma\bar{p} & n_y\gamma\bar{p} & \bar{w}\end{pmatrix} \tag{4}$$

with $\bar{w}=n_x\tilde{u}+n_y\tilde{v}$, $\quad \tilde{w}=n_x\tilde{u}+n_y\tilde{v}$.

It is easy to verify that the matrix $\tilde{A}$ has real eigenvalues and a complete set of eigenvectors.

For $n_x^2+n_y^2=1$, the eigenvalues are given by:

$$\lambda_1=\bar{w}, \quad \lambda_2=\bar{w}, \quad \lambda_3=\tilde{w}+\bar{c} \quad \textit{and} \quad \lambda_4=\tilde{w}-\bar{c} \tag{5}$$

where $\tilde{w}=(\bar{w}+\tilde{w})/2$, $\bar{c}^2=\gamma\bar{p}/\bar{\rho}+(\bar{w}-\tilde{w})^2/4$.

The matrix $\tilde{A}$ can be split into positive and negative parts by:

$$\tilde{A}^+=R\Lambda^+L, \quad \tilde{A}=R\Lambda^-L$$

where $R$ and $L$ denote right and left eigenvector matrices, in orthonormal form and where

$$\Lambda^+=\mathrm{diag}(\lambda_1^+,\lambda_2^+,\lambda_3^+,\lambda_4^+), \quad \Lambda^-=\mathrm{diag}(\lambda_1^-,\lambda_2^-,\lambda_3^-,\lambda_4^-)$$

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

Positive and negative matrices are matrices with non-negative and non-positive eigenvalues respectively.

The matrix $T$ is the transformation matrix between differences of conservative variables and differences of primitive variables, i.e.

$$\Delta\zeta=T\Delta\xi \tag{6}$$

where $\zeta$ is the vector of conservative variables

$$\zeta^T=\{\rho,\rho u,\rho v,\rho E\}$$

with the total energy

$$E=p/(\gamma-1)\rho+\tfrac{1}{2}u^2+\tfrac{1}{2}v^2$$

A combination of (3) and (6) gives:

$$\Delta\phi = T\tilde{A}T^{-1}\Delta\zeta = TR\Lambda LT^{-1}\Delta\zeta = A\Delta\zeta \tag{7}$$

Furthermore, we have

$$A^{+} = TR\Lambda^{+}LT^{-1}, \qquad A^{-} = TR\Lambda^{-}LT^{-1} \tag{8}$$

This allows a splitting of the flux-difference (3) by:

$$\Delta\phi = A^{+}\Delta\zeta + A^{-}\Delta\zeta \tag{9}$$

## VERTEX-CENTRED FINITE VOLUME FORMULATION

*Figure 1* shows the control volume centred around the node $(i,j)$. With piecewise constant interpolation of variables, the flux-difference over the surface $S_{i+\frac{1}{2}}$ of the control volume can be written as:

$$\begin{aligned}\Delta F_{i,i+1} &= \Delta y_{i+\frac{1}{2}}\Delta f_{i,i+1} + \Delta x_{i+\frac{1}{2}}\Delta g_{i,i+1} \\ &= \Delta s_{i+\frac{1}{2}}(n_x\Delta f_{i,i+1} + n_y\Delta g_{i,i+1})\end{aligned} \tag{10}$$

where $\Delta s_{i+\frac{1}{2}}$ is the length of the surface and where $n_x$ and $n_y$ denote the components of the unit outward normal to the control surface.

With the notation of the previous section, the flux-difference is:

$$\Delta F_{i,i+1} = F_{i+1} - F_i = \Delta s_{i+\frac{1}{2}} A_{i,i+1}\Delta\zeta_{i,i+1} \tag{11}$$

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:

$$|\Delta F_{i,i+1}| = \Delta s_{i+\frac{1}{2}}(A^{+}_{i,i+1} - A^{-}_{i,i+1})\Delta\zeta_{i,i+1} \tag{12}$$

Based on (12), an upwind definition of the flux is:

$$F_{i+\frac{1}{2}} = \tfrac{1}{2}[F_i + F_{i+1} - |\Delta F_{i,i+1}|] \tag{13}$$

The fact that this represents an upwind flux can be verified by writing (13) in either of the two following, equivalent, ways,

$$\begin{aligned}F_{i+\frac{1}{2}} &= F_i + \tfrac{1}{2}\Delta F_{i,i+1} - \tfrac{1}{2}|\Delta F_{i,i+1}| \\ &= F_i + \Delta s_{i+\frac{1}{2}} A^{-}_{i,i+1}\Delta\zeta_{i,i+1}\end{aligned} \tag{14}$$

$$\begin{aligned}F_{i+\frac{1}{2}} &= F_{i+1} - \tfrac{1}{2}\Delta F_{i,i+1} - \tfrac{1}{2}|\Delta F_{i,i+1}| \\ &= F_{i+1} - \Delta s_{i+\frac{1}{2}} A^{+}_{i,i+1}\Delta\zeta_{i,i+1}\end{aligned} \tag{15}$$

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

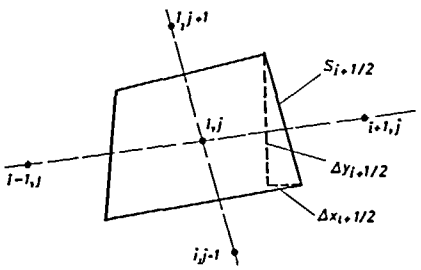


*Figure 1* Control volume around node $(i,j)$

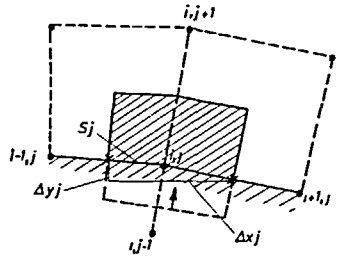


*Figure 2* Control volume at solid boundary

$A_{i,i+1}$ 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. With (14) and (15), the flux balance on the control volume of *Figure 1* has the form (omitting non-varying indices):

$$\Delta s_{i+\frac{1}{2}} A^-_{i,i+1}[\zeta_{i+1}-\zeta_i]+\Delta s_{i-\frac{1}{2}} A^+_{i,i-1}[\zeta_i-\zeta_{i-1}]+ \Delta s_{j+\frac{1}{2}} A^-_{j,j+1}[\zeta_{j+1}-\zeta_j]+\Delta s_{j-\frac{1}{2}} A^+_{j,j-1}[\zeta_j-\zeta_{j-1}]=0 \quad (16)$$

The set formed by (16) for all nodes is a so-called positive set. This can be seen by writing (16) as:

$$C\zeta_{i,j}=\Delta s_{i-\frac{1}{2}} A^+_{i,i-1}\zeta_{i-1}+\Delta s_{i+\frac{1}{2}}(-A^-_{i,i+1})\zeta_{i+1}+ \Delta s_{j-\frac{1}{2}} A^+_{j,j-1}\zeta_{j-1}+\Delta s_{j+\frac{1}{2}}(-A^-_{j,j+1})\zeta_{j+1} \quad (17)$$

where $C$ is the sum of the matrix coefficients in the right hand side and where these coefficients have non-negative 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 at each node, all components of the vector of dependent variables $\zeta$ are relaxed simultaneously.

In practice, the flux-balance (16) is formed by summing expressions of type (14) over all surfaces, using the appropriate components of the unit outgoing normal $n_x$ and $n_y$ in the definition of the Jacobian (7).

## BOUNDARY CONDITIONS

*Figure 2* shows the half-volume centred around a node on a solid boundary. This half-volume can be seen as the limit of a complete volume in which one of the sides tends to the boundary. As a consequence, the flux on the side $S_j$ of the boundary control volume can be expressed, according to (15), by:

$$F_j-\Delta s_j A^+_{i,j}(\zeta_j-\zeta_{j+1}) \quad (18)$$

where the matrix $A_{i,j}$ is calculated at the node $(i,j)$.

With the definition (18), the flux balance on the control volume at the boundary takes the form (16) in which a node outside the domain is also included. This node, however, can be eliminated. On a solid boundary, two eigenvalues of the matrix $A_{i,j}$ are zero due to the boundary condition of tangentiality:

$$\lambda_1=\bar{w}=0, \qquad \lambda_2=\bar{w}=0$$

The third and the fourth eigenvalues are given by:

$$\lambda_3=\bar{c}, \qquad \lambda_4=-\bar{c}$$

As a consequence, the rank of the matrix $A^+_{i,j}$ is equal to one. This means that premultiplication of the flux-balance on a solid boundary by a left eigenvector associated to a zero eigenvalue of $A^+_{i,j}$, yields an equation where the outside node is eliminated. This results in three equations, which are supplemented by the boundary condition of tangentiality. A similar procedure can be used at inflow and at outflow. At subsonic inflow, one equation is obtained and three additional relations are determined from the boundary conditions. At subsonic outflow, three equations are obtained and one boundary condition is to be given. Physically, at inlet stagnation pressure, stagnation temperature and flow direction are to be prescribed. At outlet, Mach number can be prescribed. Explicit expressions of the combinations of the equations at boundaries are given in Reference 1.

Due to the linearity of the condition of impermeability, the set of equations on a solid boundary is a quasi-linear set which is similar to the set in the flow field. At inflow and outflow boundaries, the physical boundary conditions are highly non-linear combinations of the dependent variables.

Therefore, the introduction of the boundary conditions at inlet and outlet, in the way as described above, necessitates iteration which is a further complication. Therefore, it is better to treat the nodes at inlet and outlet as auxiliary points and to determine the variables at these points by extrapolation. At inlet, the Mach number is extrapolated along streamlines. Together with the given boundary conditions, this determines all flow variables directly. 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 directly.

## SECOND ORDER FORMULATION

In order to obtain second order accuracy, the definition of the flux (13) is to be modified. First, we remark that, using (7), the flux-difference (11) can be written as:

$$\Delta F_{i,i+1} = \Delta s_{i+\frac{1}{2}} \sum_n r^n_{i+\frac{1}{2}} \lambda^n_{i+\frac{1}{2}} l^n_{i+\frac{1}{2}} \Delta\zeta_{i,i+1} \tag{19}$$

where the superscript $n$ refers to the $n$th eigenvalue and where $r^n$ and $l^n$ denote the $n$th right and left eigenvectors. $r^n$ and $l^n$ are components of $TR$ and $LT^{-1}$. By denoting the projection of $\Delta\zeta_{i,i+1}$ on the $n$th eigenvector by:

$$\sigma^n_{i+\frac{1}{2}} = l^n_{i+\frac{1}{2}} \Delta\zeta_{i,i+1}$$

(19) can be written as:

$$\Delta F_{i,i+1} = \sum_n \Delta F^n_{i,i+1} = \Delta s_{i+\frac{1}{2}} \sum_n r^n_{i+\frac{1}{2}} \lambda^n_{i+\frac{1}{2}} \sigma^n_{i+\frac{1}{2}} = \Delta s_{i+\frac{1}{2}} \sum_n r^n_{i+\frac{1}{2}} \tau^n_{i+\frac{1}{2}} \tag{20}$$

where $\Delta F^n_{i,i+1}$ is the component of the flux-difference associated to the $n$th eigenvalue and $\tau^n_{i+\frac{1}{2}}$ is the projection of the flux-difference on the $n$th eigenvector.

Using (20), the first order flux (13) can be written as:

$$F_{i+\frac{1}{2}} = \tfrac{1}{2}(F_i + F_{i+1}) - \tfrac{1}{2} \sum_n \Delta F^{n+}_{i,i+1} + \tfrac{1}{2} \sum_n \Delta F^{n-}_{i,i+1} \tag{21}$$

where the + and − superscripts denote the positive and negative parts of the components of the flux-difference, i.e. the parts obtained by taking the positive and negative parts of the eigenvalues. According to Chakravarthy and Osher[11], assuming a structured sufficiently smooth grid, a second order flux corresponding to (21) can be defined by:

$$F_{i+\frac{1}{2}} = \tfrac{1}{2}(F_i + F_{i+1}) - \tfrac{1}{2} \sum_n \Delta F^{n+}_{i,i+1} + \tfrac{1}{2} \sum_n \Delta F^{n-}_{i,i+1} + \tfrac{1}{2} \sum_n \tilde{\Delta} F^{n+}_{i-1,i} - \tfrac{1}{2} \sum_n \tilde{\Delta} F^{n-}_{i+1,i+2} \tag{22}$$

where

$$\begin{aligned} \tilde{\Delta} F^{n+}_{i-1,i} &= \Delta s_{i+\frac{1}{2}} r^n_{i+\frac{1}{2}} \lambda^n_{i+\frac{1}{2}} l^n_{i+\frac{1}{2}} \Delta\zeta_{i-1,i} \\ &= \Delta s_{i+\frac{1}{2}} r^n_{i+\frac{1}{2}} \lambda^n_{i+\frac{1}{2}} \tilde{\sigma}^n_{i-\frac{1}{2}} \end{aligned} \tag{23}$$

with a similar definition for $\tilde{\Delta} F^{n-}_{i+1,i+2}$.

Clearly (23) is constructed by considering a flux-difference over the surface $S_{i+\frac{1}{2}}$, i.e. using the geometry of this surface, with data shifted in the negative $i$-direction. $\tilde{\sigma}^n_{i-\frac{1}{2}}$ represents the projection of the shifted difference of the dependent variables on the $n$th eigenvector of the original flux-difference. The second order correction could also be defined using the $\tau$-variables, i.e. the projections of the flux-difference. This would mean that the eigenvalue in (23) is also shifted. In practice, there is little difference between the results of both formulations. In the sequel we use (23) only.

The definition (22) corresponds to a second order upwind flux. This can be clearly seen by considering the case where all eigenvalues have the same sign. Second order accuracy can also be achieved by taking a central definition of the flux vector:

$$\bar{F}_{i+\frac{1}{2}} = \tfrac{1}{2}(F_i + F_{i+1}) \tag{24}$$

As is well known, using either (22) or (24) leads to a scheme which is not monotonicity preserving so that wiggles in the solution become possible. Following the theory of the flux limiters[12], a combination of (22) and (24) is to be taken. This has the form:

$$F_{i+\frac{1}{2}}=\tfrac{1}{2}(F_i+F_{i+1})-\tfrac{1}{2}\sum_n \Delta F^{n+}_{i,i+1}+\tfrac{1}{2}\sum_n \Delta F^{n-}_{i,i+1}+\tfrac{1}{2}\sum_n \tilde{\tilde{\Delta}} F^{n+}_{i-1,i}-\tfrac{1}{2}\sum_n \tilde{\tilde{\Delta}} F^{n-}_{i+1,i+2} \tag{25}$$

with

$$\tilde{\tilde{\Delta}} F^{n+}_{i-1,i}=\mathrm{Lim}(\tilde{\Delta} F^{n+}_{i-1,i}, \Delta F^{n+}_{i,i+1}) \tag{26}$$

$$\tilde{\tilde{\Delta}} F^{n-}_{i+1,i+2}=\mathrm{Lim}(\tilde{\Delta} F^{n-}_{i+1,i+2}, \Delta F^{n-}_{i,i+1}) \tag{27}$$

where Lim denotes some limited combination of both arguments. We choose here the simplest possible form of a limiter, i.e. Lim = MinMod, where the function MinMod returns the argument with minimum absolute value if both arguments have the same sign and returns zero otherwise. Using the limiter to the vectors (26), (27) means that the limiter is used per $\sigma$-component.

In the vicinity of boundaries, some components of flux-differences in (26) or (27) do not exist. For these components, the limiter then returns a zero. This does not degrade the second order accuracy since, due to the characteristic boundary treatment, these components do not enter the boundary equations. The foregoing second order correction procedure is called the flux-extrapolation technique. In contrast to the more common MUSCL-technique[5–8], it gives the second order correction in an explicit way. Flux-extrapolation is therefore much simpler to use with defect-correction.

## MULTIGRID DEFECT-CORRECTION FORMULATION

Since, for the discretization obtained by the second order formulation, the positivity is not guaranteed, a relaxation solution is impossible. Therefore, following Hemker[5], a defect-correction formulation is used as solution procedure. By denoting, symbolically, the first order and second order formulation on the finest grid by:

$$L^1_h = r^1_h \tag{28}$$

$$L^2_h = r^2_h \tag{29}$$

a defect correction means that (28) is replaced by:

$$L^1_h = r^1_h + [(L^1_h - r^1_h) - (L^2_h - r^2_h)] \tag{30}$$

where $L$ and $r$ indicate left and right hand sides.

In (30) the difference of the defects of the first and second order discretization is added to the right hand side of the first order system. This difference is called the defect-correction. A multigrid method based on the first order system is used in which for every visit to the finest grid the defect-correction is recalculated using the newest values of the variables. *Figure 3* shows the cycle-structure of the multigrid method. Both the starting cycle and the repeated cycle have a W-form. A full approximation scheme (FAS) on the non-linear equations (16) is used. The relaxation algorithm is Gauss–Seidel in lexicographic order, but alternately starting in the lower left corner going up in $j$-direction and in the upper left corner going down in $j$-direction. Three (or four) relaxations are done per level. In relaxing the set of (16), the coefficients are formed with the latest available information. This means, for instance, that the coefficient $A^+_{i,i-1}$ 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 undertaken. This means that the set of (16) is treated as a quasi-linear set and that the multigrid procedure corresponds to a Picard iteration and not to a Newton iteration. As restriction operator for residuals, full weighting in the flow field and injection at the boundaries is used. The restriction for function values is injection and the prolongation operator is bilinear interpolation.

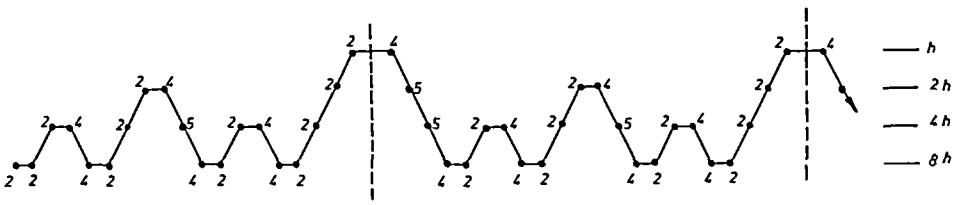


*Figure 3* The multigrid cycle

In *Figure* 3, the operation count is indicated for three relaxations per level. A relaxation on the current grid is taken as one local work unit. A residual evaluation plus the associated grid transfer is also taken as one local work unit. Hence, the 5 in *Figure 3*, in going down, stands for the construction of the right hand side in the FAS formulation, three relaxations and one residual evaluation. With this way of evaluating the work, the cost of the repeated cycle is 13.0625 work units on the finest level. The cost of the starting cycle is 7.5 work units. For four relaxations per level, this is respectively 16.6875 and 9.8125 work units.

## COMPUTATIONAL EXAMPLES

*Figure 4* shows the GAMM-test problem for transonic flows[13]. The finest grid has 97 × 33 nodes. The Figure shows every other gridline. The circular perturbation of the channel has a height of 4.2% of the chord and the height of the channel is 2.073 times the chord. The incoming flow is uniform and the Mach number of the flow at outlet is prescribed to be 0.85.

*Figures 5* and *6* show the first order and second order iso-Machline results. There is almost no difference between both results. This can be explained by the alignment of the shock with the gridlines.

*Figure 7* shows the convergence behaviour of the first order and second order multigrid methods. The calculation starts from a uniform flow with Mach number 0.85 on the coarsest grid and three relaxations are done per level with a relaxation factor of 0.9. The residual shown is the maximum over all equations and all nodes after normalizing the equations. The normalization process means that all variables are divided by their value at inflow and that the smallest surface of the smallest cell in the flow field is set equal to 0.5 (the value 1 seems not to be fair). For the second order method, the defect-correction is applied from the first cycle. The residual reduction in the first order method is about 0.927 per work unit. For the second order formulation it is about 0.944 per work unit in the first phase of the convergence (up to about 120 work units). The residual reduction per cycle for the second order method is about 0.444. This is not an excellent but is an acceptable multigrid performance. Some saturation occurs after the first phase of the convergence, which is typical for a defect-correction procedure.

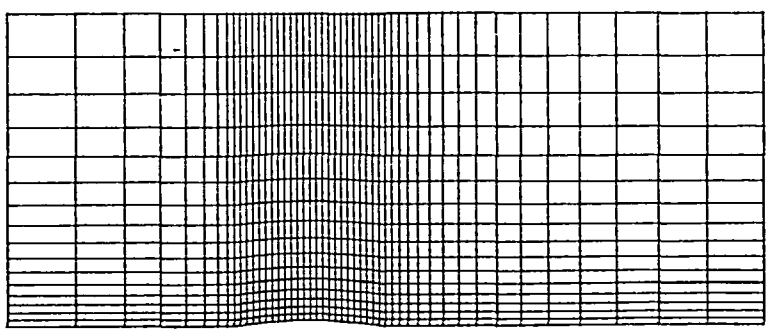

*Figure 4* The GAMM-test problem for transonic flows

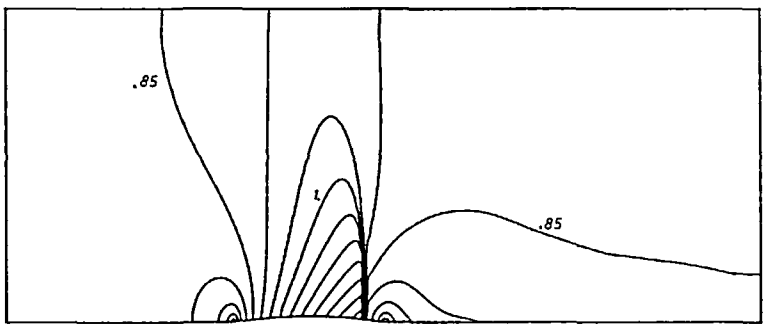


*Figure 5* First order solution for the GAMM-test problem; iso-Machlines per 0.05

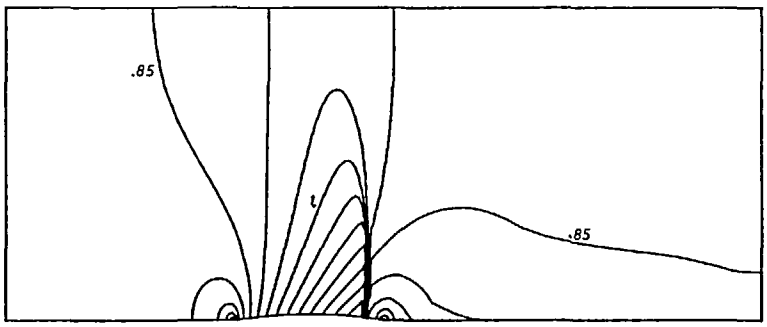


*Figure 6* Second order solution for the GAMM-test problem; iso-Machlines per 0.05

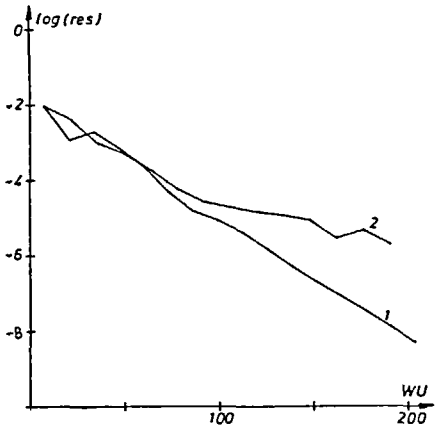


*Figure 7* Convergence behaviour of first order (1) and second order (2) solutions for the GAMM-test problem

In practice, this saturation has not much meaning since the result obtained at a convergence level of $10^{-4}$ cannot be distinguished from the fully converged solution.

*Figure 8* shows a test problem as used by Koren and Spekreijse[8]. Again the grid has $97 \times 33$ nodes and the Figure shows every other gridline. The circular perturbation of the channel has a height of 4% of the chord and the height of the channel is equal to the chord. The oncoming flow is uniform with Mach number 1.40.

*Figures 9* and *10* contain plots of the first order and the second order iso-Machline results and show that the difference between both solutions is very large. In the first order solution,

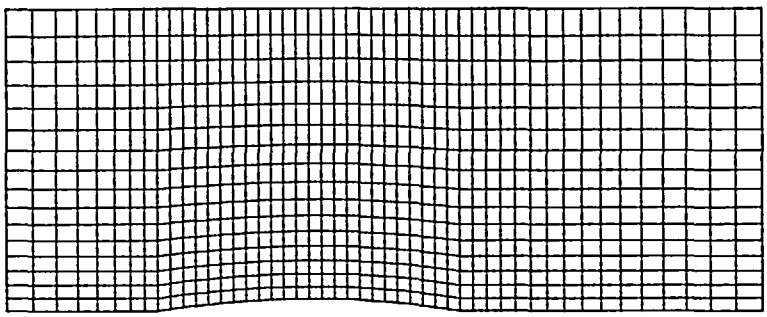

*Figure 8* The test problem of Koren and Spekreijse (K & S)

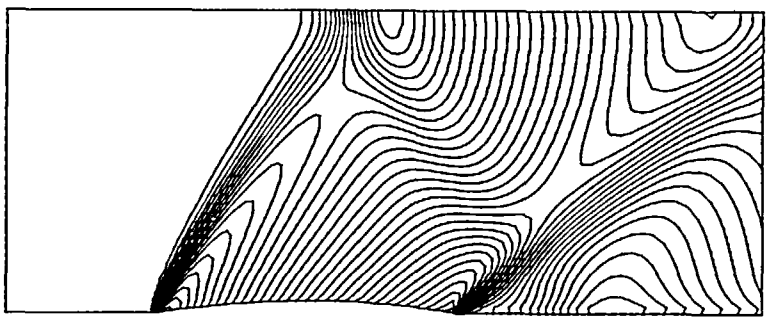

*Figure 9* First order solution for the K & S-test problem; iso-Machlines per 0.02; most left iso-Machline = 1.40

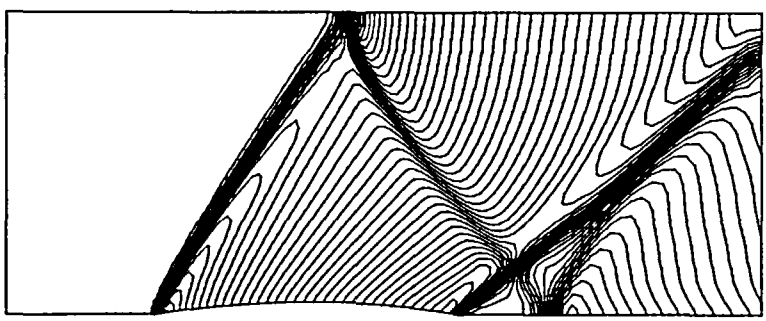

*Figure 10* Second order solution for the K & S-test problem; iso-Machlines per 0.02; most left iso-Machline = 1.40

the interaction between the reflected shock wave and the shock wave coming from the trailing edge of the circular perturbation cannot be seen. This interaction is clearly visible in the second order solution. This test problem is thus much more critical than the GAMM-test problem.

*Figure 11* shows the convergence behaviour of the first order and the second order multigrid methods. The procedure is identical to the one for the first test case, apart from the relaxation factor. For this supersonic problem, this factor is set equal to 1. The residual reduction in the first order method is about 0.800 per work unit which degrades to about 0.970 per work unit for the second order method. This is 0.675 per cycle. The difference in performance is now much larger between first and second order formulations due to the large difference in solutions.

The accuracy obtained for this test problem is comparable to the accuracy obtained by Koren and Spekreijse[8]. The convergence of the method used here is better in terms of work units. This better performance is obtained despite the non-differentiability of the splitting technique used. Moreover, the work unit here is much cheaper due to the simplicity of the splitting technique.


*Figure 11* Convergence behaviour of first order (1) and second order (2) solutions for the K & S-test problem

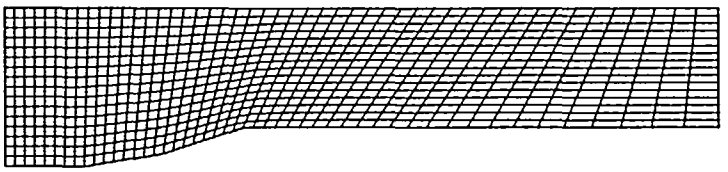

*Figure 12* The step-test problem

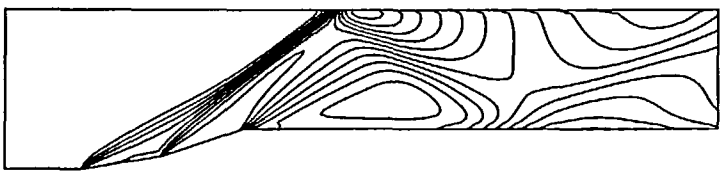

*Figure 13* First order solution for the step-test problem; iso-Machlines per 0.1; most left iso-Machline = 2.90

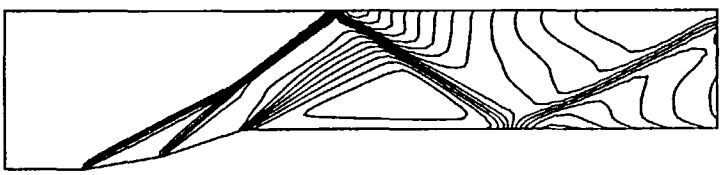

*Figure 14* Second order solution for the step-test problem; iso-Machlines per 0.1; most left iso-Machline = 2.90

*Figure 12* shows a third test problem. Again the grid has 97 × 33 nodes and every other gridline is shown. The channel has a more or less gradual step and the oncoming flow is uniform with Mach number 2.9.

*Figures 13* and *14* show the first order and second order iso-Machline results. The conclusion with respect to accuracy is more or less the same as in the second test case.

The calculation starts from a uniform flow with Mach number 2.9 on the coarsest grid. This test problem is somewhat more difficult than the previous test problems. To obtain reasonable convergence, four iterations per level are now necessary and the defect-correction cannot be started from the first cycle. *Figure 15* shows the convergence behaviour for the first order method and the second order method with defect-correction used after the second full cycle. The residual reduction of the second order method is 0.947 per work unit, or 0.400 per cycle. The residual reduction on itself is good but there is an important delay in convergence due to the jump in

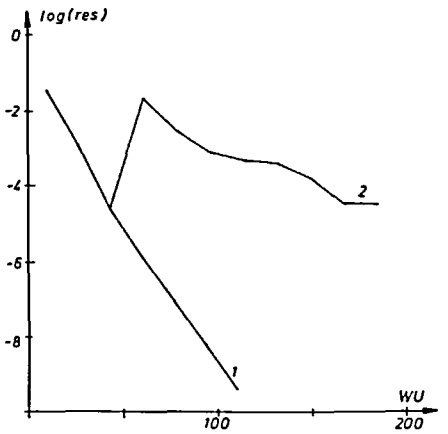


*Figure 15* Convergence behaviour of first order (1) and second order (2) solutions for the step test problem

residual after switching on the defect-correction. Overall the performance is acceptable and a residual of $10^{-4}$ is reached after about 150 work units.

## CONCLUSION

It has been shown that using a very simple flux-difference splitting technique as a basis for a relaxation type multigrid method, accuracy and convergence speed can be comparable to what is obtained with more common, more complicated techniques.

## ACKNOWLEDGEMENT

The research reported in this paper was granted by the Belgian National Science Foundation (N.F.W.O.).

## REFERENCES

1 Dick, E. A flux-difference splitting method for steady Euler equations, *J. Comp. Phys.*, **76**, 19–32 (1988)
2 Roe, P. L. Approximate Riemann solvers, parameter vectors and difference schemes, *J. Comp. Phys.*, **43**, 357–372 (1981)
3 Lombard, C. K., Oliger, J. and Yang, J. Y. A natural conservative flux difference splitting for the hyperbolic systems of gas dynamics, *AIAA paper 82-0976* (1982)
4 Osher, S. and Chakravarthy, S. R. Upwind schemes and boundary conditions with applications to Euler equations in general geometries, *J. Comp. Phys.*, **50**, 477–481 (1983)
5 Hemker, P. W. Defect correction and higher order schemes for the multigrid solution of the steady Euler equations, *Lecture Notes Math.*, **1228**, 149–165 (1986)
6 Hemker, P. W. and Spekreijse, S. Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations, *Appl. Num. Math.*, **2**, 475–493 (1986)
7 Koren, B. Defect correction and multigrid for an efficient and accurate computation of airfoil flows, *J. Comp. Phys.*, **77**, 183–206 (1988)
8 Koren, B. and Spekreijse, S. Solution of the steady Euler equations by a multigrid method, *Lecture Notes Pure Appl. Math.*, **110**, 323–336 (1988)
9 Dick, E. A multigrid flux-difference splitting method for steady Euler equations, *Proc. 4th Copper Mountain Conf. Multigrid Methods*, SIAM, pp. 117–129 (1989)
10 Dick, E. Multigrid formulation of polynomial flux-difference splitting for steady Euler equations, *J. Comp. Phys.*, **91**, 161–173 (1990)
11 Chakravarthy, S. R. and Osher, S. A new class of high accuracy TVD schemes for hyperbolic conservation laws, *AIAA paper 85-0363* (1985)
12 Sweby, P. K. High resolution schemes using flux limiters for hyperbolic conservation laws, *SIAM J. Num. Anal.*, **21**, 995–1011 (1984)
13 Rizzi, A. and Viviand, H. (eds.), Numerical methods for the computation of inviscid transonic flows with shock waves, *Notes Num. Fluid Dynamics* Vol. 3, Vieweg, Braunschweig (1981)