AIAA JOURNAL
Vol. 33, No. 5, May 1995

Upwind-Biased Finite-Volume Technique Solving Navier–Stokes Equations on Irregular Meshes

J. A. Essers,* M. Delanaye,† and P. Rogiest†
University of Liège, B-4000 Liege, Belgium

New node-centered finite-volume discretizations of advective and diffusive derivatives on structured meshes with quadrilateral cells are presented. They are applied to the solution of Euler and full Navier–Stokes equations using a pseudotime-dependent approach. The advective derivatives are split and upwind biased. The most interesting aspect of the scheme lies in its ability to provide good accuracy on meshes with severe skewness and stretching distortions. A very sensitive detector is presented which is capable of selectively identifying shock waves and insufficiently resolved shear layers. It is used to automatically switch to a first-order upwind scheme in some regions and provides a sharp and monotone shock capture. Results obtained for plane and axisymmetric steady supersonic laminar flows are discussed. They include blunt-body flows, a shock/boundary-layer interaction, and a flow over a compression corner.

I. Introduction

MOST classical finite-volume schemes for the solution of the Euler and Navier–Stokes equations require very regular and smooth meshes. Severe mesh skewness and stretching distortions can strongly deteriorate the accuracy and lead to significant parasitic diffusion or antidiffusion effects. These effects can be responsible for relatively poor evaluations of important flow properties and eventually for numerical instabilities. The accuracy of several space-centered schemes using structured meshes has been theoretically and practically evaluated by Renard and Essers.[1] For that purpose, they considered a simple model governed by a linear advection-diffusion equation. Starting from an unstretched rectangular mesh, more and more severe distortions were introduced in the mesh using a special random grid distortion procedure. Different families of structured grids, each with different amounts of distortions measured by a parameter ranging from 0 to 1 were thus created and used to solve the model problem. Errors obtained with finer and finer grids presenting the same amount of distortion were evaluated. All of the tested schemes, which are frequently thought of as being second-order accurate, were found zeroth-order accurate, i.e., inconsistent, for diffusive derivatives. For the advective derivatives, the best schemes were found first-order accurate only, whereas many others were inconsistent.

The main aim of this paper is to present new upwind-biased finite-volume schemes that do not exhibit the parasitic numerical effects appearing in other schemes and can, therefore, lead to a significantly improved accuracy on irregular meshes. They have been used to solve the steady Euler and Navier–Stokes equations for plane and axisymmetric supersonic and hypersonic flows using structured, distorted, node-centered meshes. Solutions are obtained from a pseudounsteady approach using a multistep Runge–Kutta scheme with variable time stepping. Our code simulates laminar perfect gas flows with a constant Prandtl number. The variable viscosity is given by the Sutherland law modified for temperatures below 120 K (see, e.g., Ref. 2 for more details). As real gas effects are not accounted for, some results are physically unrealistic but can be useful for comparisons with other codes.

Received June 25, 1993; presented as Paper 93-3377 at the AIAA 11th Computational Fluid Dynamics Conference, Orlando, FL, July 6–9, 1993; revision received May 13, 1994; accepted for publication June 17, 1994.

*Professor, Head of Aerodynamics Group, Institute of Mechanics and Aeronautics. Member AIAA.

†Research Assistant, Aerodynamics Group, Institute of Mechanics and Aeronautics.

Essers and Renard[3] used a simplified version of the scheme presented in this paper to compute a shockless transonic nozzle flow using a smooth and a severely distorted mesh. Comparative calculations were performed with different classical finite-volume schemes including a scheme using second-order upwinding. For the smooth mesh, all schemes lead to similar results. For the distorted mesh, the results obtained with the new scheme remained almost unchanged, whereas the accuracy obtained with the classical schemes became quite poor. In particular, severe oscillations appeared in the Mach number and total pressure distributions computed with the second-order upwind scheme.

Recently, Barth[4] proposed some high-order reconstruction techniques that can also be used on irregular meshes. The order of accuracy of our scheme, which is also based on a quadratic extrapolation of conservative variables on the control volume sides, is similar to that obtained with Barth's quadratic reconstruction. However, our scheme is different and presumably less time consuming. Since Barth did not present results for viscous supersonic blunt body flows, a direct comparison of the two schemes was unfortunately not possible. A comparison with results[2] obtained by Vankeirsbilck and Deconinck, who also use polynomial reconstruction, is presented in this paper.

II. Space-Centered Discretizations of First- and Second-Order Derivatives

We have to discretize first- and second-order space derivatives on eventually distorted structured grids. In the following, each node is identified by the subscripts (m, j). The centered discretized forms of the derivatives at any node (m, j) should not couple more than nine mesh points, i.e., (m, j) itself and its closest neighbors. They should unconditionally meet the following accuracy requirements: second- (first-) order accuracy at least for all first- (second-) order derivatives; and discretized second-order derivatives should become second-order accurate on unstretched rectangular meshes. Discretization operators just mentioned are essentially used to construct appropriate interpolation/extrapolation formulas providing sufficiently accurate evaluations of flux vectors at the center of each control volume side. As a result, those discretizations do not have to be conservative.

Let us denote by $\partial_i v$ $(i = 1, \ldots, 5)$ the exact value of first- and second-order derivatives, with respect to independent variables x and y, of any function v defined at all nodal points ($i = 1, 2$ corresponding to first-order derivatives). Discretized forms of these derivatives at mesh point (m, j) meeting all aforementioned requirements are denoted by $L_i^{(m,j)}[v]$, or more simply by L_i. These discrete operators can be defined in various different ways which do not all lead to equivalent formulas. We present a very systematic procedure to construct those discretizations. More details on that

procedure are given by Essers and Renard.[3] We first use a classical finite-volume scheme to construct space-centered operators $L_i^*[\nu]$. Using subscript l for a local numbering of the mesh points, these operators appear as linear combinations of the ν_l at mesh point (m, j) itself ($l = 1$) and at its eight closest neighbors ($l = 2, \ldots, 9$)

$$L_i^{*(m,j)}[\nu] = \sum_{l=2}^{9} e_{il}^{*(m,j)}(\nu_l - \nu_1) \tag{1}$$

where the $e_{il}^{*(m,j)}$ are known coefficients. A local Taylor series expression of ν_l gives

$$\nu_l - \nu_1 = \sum_{k=1}^{5} d_k^{(m,j)}(\partial_k \nu)_{m,j} + \text{HOT} \tag{2}$$

where $d_1 = x_l - x_1$, $d_2 = y_l - y_1$, $d_3 = 0.5(d_1)^2$, $d_4 = 0.5(d_2)^2$, $d_5 = d_1 d_2$, and HOT denotes higher order terms. Substituting Eq. (2) into Eq. (1) yields

$$L_i^{*(m,j)}[\nu] = \sum_{k=1}^{5} C_{ik}^{(m,j)}(\partial_k \nu)_{m,j} + \text{HOT} \tag{3}$$

The 5×5 coefficients defining the two-dimensional array $C^{(m,j)}$ can be computed as follows:

$$C_{ik}^{(m,j)} = \sum_{l=2}^{9} e_{il}^{*(m,j)} d_k^{(m,j)} \tag{4}$$

Note that because of consistency requirements, all C_{ik} corresponding to $i = 3, 4, 5$ and $k = 1, 2$ are equal to zero. Corrected operators, which can be proved to exhibit the appropriate accuracy, are then constructed as follows:

$$L_i^{(m,j)} = \sum_{k=1}^{5} D_{ik}^{(m,j)} L_k^{*(m,j)} \tag{5}$$

where $D^{(m,j)}$ is the inverse of array $C^{(m,j)}$. That concise formula actually means that the corrected operators appear as linear combinations similar to Eq. (1) but using modified coefficients defined by

$$e_{il}^{(m,j)} = \sum_{k=1}^{5} D_{ik}^{(m,j)} e_{kl}^{*(m,j)} \tag{6}$$

A similar procedure is used to define corrected operators at boundary nodes. They couple three boundary mesh points and six points located inside the domain on two different mesh lines.

As nine coefficients corresponding to the weight of the neighboring mesh points are required to define each corrected operator, 9×5 coefficients should be evaluated at each mesh point. For fixed grids, they are computed only once and stored in memory at the beginning of the calculation.

III. Accuracy of Finite-Volume Discretizations of the Navier–Stokes Equations

The two-dimensional time-dependent Navier–Stokes equations can be written in the following conservative form:

$$\partial_t \boldsymbol{s} + S_a[\boldsymbol{f}, \boldsymbol{g}] + S_d[\boldsymbol{q}, \boldsymbol{r}] = 0 \tag{7}$$

where $\boldsymbol{s}$ is the vector of flow conservative dependent variables, $S_a \equiv \partial_x f + \partial_y g$ is the advective operator, $\boldsymbol{f}(\boldsymbol{s})$ and $\boldsymbol{g}(\boldsymbol{s})$ denote the advective flux vectors, $S_d \equiv \partial_x q + \partial_y r$ is the diffusive operator, and diffusive flux vectors $\boldsymbol{q}(w, w_x, w_y)$ and $\boldsymbol{r}(w, w_x, w_y)$ are expressed in terms of primitive flow variables w and their first-order derivatives.

Conservative flow variables are computed at nodal points (m, j) from a finite-volume space discretization of Eq. (7) using quadrilateral control volumes. The latter are limited by straight lines joining the centers of successive mesh cells (Fig. 1). Points located at the center of each control volume side play an essential role in the description of our scheme. They will be referred to as control points.

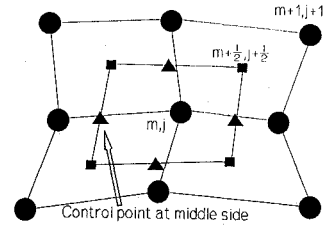


Fig. 1 Control points and control volumes.

The aforementioned discretized form of Eq. (7) is written as follows at point (m, j):

$$\partial_t s_{m,j} + S_a^{(m,j)} + S_d^{(m,j)} = 0 \tag{8a}$$

with

$$S_a^{(m,j)} = 1/A_{m,j}\Big[F_{(m+\frac{1}{2},j)} - F_{(m-\frac{1}{2},j)} + G_{(m,j+\frac{1}{2})} - G_{(m,j-\frac{1}{2})}\Big] \tag{8b}$$

$$S_d^{(m,j)} = 1/A_{m,j}\Big[Q_{(m+\frac{1}{2},j)} - Q_{(m-\frac{1}{2},j)} + R_{(m,j+\frac{1}{2})} - R_{(m,j-\frac{1}{2})}\Big] \tag{8c}$$

where A is the area of the control volume, and F and G denote approximations of the integral $\int(\boldsymbol{g}\,dy - \boldsymbol{f}\,dx)$ taken over the appropriate control volume sides.

To present the scheme actually used to discretize the advective fluxes, it is first useful to consider an unrealistic scheme which assumes that the exact values $\bar{\boldsymbol{s}}$ of $\boldsymbol{s}$ are known at all control points. If this statement was true, the flux integrals could obviously be approximated as follows:

$$F_{(m+\frac{1}{2},j)} = \boldsymbol{f}(\bar{\boldsymbol{s}})\left(y_{m+\frac{1}{2},j+\frac{1}{2}} - y_{m+\frac{1}{2},j-\frac{1}{2}}\right) - \boldsymbol{g}(\bar{\boldsymbol{s}})\left(x_{m+\frac{1}{2},j+\frac{1}{2}} - x_{m+\frac{1}{2},j-\frac{1}{2}}\right) \tag{9}$$

$(m + \frac{1}{2}, j + \frac{1}{2})$ and $(m + \frac{1}{2}, j - \frac{1}{2})$ denoting the centers of the mesh cells. Using approximations similar to Eq. (9) for other advective flux integrals, formula (8b) provides the approximate value of the advective operator. By means of truncated Taylor series expansions near (m, j) of $\boldsymbol{f}(\bar{\boldsymbol{s}})$ and $\boldsymbol{g}(\bar{\boldsymbol{s}})$ at the four control points, we can perform a theoretical analysis of the dominant discretization error of the unrealistic scheme in the evaluation of the advective operator. This leads to the following conclusions. For uniform rectangular grids, the dominant error is essentially dispersive and of second order with respect to the mesh sizes. For irregular stretched and skewed meshes, however, the scheme is first-order accurate only. The dispersive errors already mentioned can then be dominated by a diffusive or antidiffusive error term equal to the sum of the following two expressions:

$$E_{1,m,j} = a_{11}^{(m,j)}\left(\partial_{xx}^2 \boldsymbol{f}\right)_{m,j} + a_{12}^{(m,j)}\left(\partial_{xy}^2 \boldsymbol{f}\right)_{m,j} + a_{22}^{(m,j)}\left(\partial_{yy}^2 \boldsymbol{f}\right)_{(m,j)} \tag{10}$$

The coefficients a and b depend linearly on variations of the local characteristic mesh sizes. They can be computed easily.

As exact values $\bar{\boldsymbol{s}}$ of $\boldsymbol{s}$ are unknown at the control points, the unrealistic scheme is obviously unusable. In practice, the advective fluxes F and G are computed using classical splitting algorithms as functions of left and right values $\boldsymbol{s}_-$ and $\boldsymbol{s}_+$, which are both supposed to be appropriate approximations of $\boldsymbol{s}$ at the control points. Let us denote by n the order of accuracy obtained with these left and right approximations. The relative error achieved in the evaluation of the advective flux vectors on each individual control volume side is of the same order. Now assume that the contribution of each flux is introduced into Eq. (8b) to evaluate the advective operator S_a

at the nodal point (m, j) and that a local Taylor series expansion is performed near the latter to estimate the dominant error related to that evaluation. The difference between that error and the error obtained when using the unrealistic scheme can be proved to be of the order of $n-1$ only with respect to the local characteristic mesh sizes on irregular grids. From these remarks, we can conclude that for irregular grids, the order of accuracy achieved in the evaluation of the advective flux vector derivatives is $n-1$ at best but cannot be better than the accuracy obtained with the unrealistic scheme, i.e., first order. The following two essential properties can then easily be proved.

Property 1: to unconditionally ensure the consistency of the advective derivatives, the extrapolation formulas used to evaluate s_- and s_+ should be at least second-order accurate.

Property 2: if the extrapolation formulas are third-order accurate (or better), the dominant diffusive error term related to the evaluation of the advective derivatives is the same as for the unrealistic scheme and, therefore, is given by Eq. (10).

If we notice that the extrapolation formulas used in the so-called first- and second-order upwind schemes are both first-order accurate only on skewed meshes, we can conclude that these schemes, which are referred to as the RH1 and RH2 schemes in the following, are generally inconsistent.

The unrealistic scheme for the discretization of the diffusive operator S_d can be defined in the same way as for the advective operator. To construct the scheme we actually use, approximations of the diffusive flux vectors $\boldsymbol{q}$ and $\boldsymbol{r}$ are first computed at the nodes and then interpolated at the control points. These interpolated values are introduced into a formula similar to Eq. (9) to compute the diffusive fluxes used in Eq. (8c). A property similar to property 1 can be used to determine the requirements to be met to ensure the unconditional consistency in the evaluation of the diffusive operators. From that property, we can conclude that the first-order derivatives of the primitive variables required to compute the nodal values of diffusive flux vectors should be evaluated with a second-order accuracy at least. The interpolation formula should also be second-order accurate.

In our code, the derivatives of the primitive variables are computed from the corrected operators $L_1[w]$ and $L_2[w]$ defined in Sec. II.

IV. Interpolation of the Diffusive Flux Vectors at the Control Points

We now discuss the procedure used to interpolate the diffusive flux vectors at any control point from their second-order accurate approximate values at six nodes. A typical stencil used is shown in Fig. 2.

The following example formula is used to interpolate flux vector $\boldsymbol{q}$ at control point P located at the center of control volume side MN:

$$\boldsymbol{q}_P \simeq \alpha \boldsymbol{q}_E + (1-\alpha)\boldsymbol{q}_O + \beta\left[\frac{1-\mu}{d_1}(\boldsymbol{q}_B - \boldsymbol{q}_A) + \frac{\mu}{d_2}(\boldsymbol{q}_D - \boldsymbol{q}_C)\right] \quad (11)$$

That formula should be such that the interpolation is unconditionally second-order accurate for any value of parameter μ. Appropriate expressions of α and β as functions of μ and of the coordinates of P at the six nodal points are obtained from lengthy analytical formulas, which can be established from the following schematic explanations. The expression between the brackets can be considered as a linear combination of two directional derivatives of $\boldsymbol{q}$ in two different directions parallel to AB and CD, respectively. Expressing them in terms of the x and y derivatives and grouping all derivatives

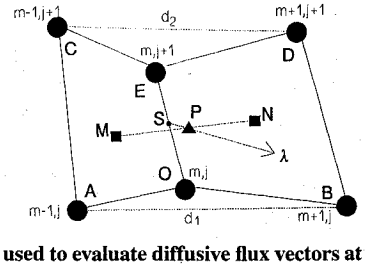


Fig. 2 Stencil used to evaluate diffusive flux vectors at a control point.

with respect to x and y, respectively, the whole expression appears as proportional to a directional derivative ∂_λ in a certain direction which can be identified. A straight line drawn from P parallel to that direction allows us to locate point S along OE, and, therefore, to compute α. The evaluation of distance SP then provides the value of β.

When some of the time-dependent Navier–Stokes equations have to be discretized at a boundary node, a part of the control volume of that node is located outside the domain. A dummy control point P then has to be used. It is generally located outside the domain and, therefore, also outside its stencil. Three nodal points of the latter are boundary nodes (points A, O, B, or C, E, D). Interpolation formula (11) is then used with a value of μ equal to 0 or 1. For all other control points, μ is chosen equal to 0.5.

When using the interpolation formulas described in this section, the diffusive flux vector derivatives evaluated at the nodal points are generally first-order accurate only, except for uniform rectangular meshes. However, that accuracy is sufficient for our purposes because the dominant discretization error is essentially dispersive and, therefore, not likely to be responsible for unexpected instabilities due to mesh distortions. Additionally, for large Reynolds numbers, all coefficients of diffusive derivatives are very small compared to those of advective derivatives.

Our viscous term discretization is not positivity preserving. However, in our calculations that property was not found to have any significant effect on the stability of the scheme, nor to be responsible for a deterioration in the accuracy of capture of shock waves.

V. Calculation of the Advective Derivatives

The left and right values of the conservative variables at all control points are evaluated with a third-order accuracy using appropriate truncated Taylor series expansions of s near the two nodal points having the control volume side in common. The derivatives appearing in these expansions are computed with the corrected centered schemes defined in Sec. II. For example, the left value at point $(m+1/2, j)$ is obtained from Eq. (12), which can be considered as a complete quadratic extrapolation

$$s_- = s_{m,j} + \sum_{i=1}^{5} \Delta_i L_i^{(m,j)}[s] \quad (12)$$

where

$$\Delta_1 = x_{(m+\frac{1}{2},j)} - x_{(m,j)}; \qquad \Delta_2 = y_{(m+\frac{1}{2},j)} - y_{(m,j)}$$

$$\Delta_3 = 0.5(\Delta_1)^2; \qquad \Delta_4 = 0.5(\Delta_2)^2; \qquad \Delta_5 = \Delta_1\Delta_2$$

The advective fluxes can then be evaluated using an appropriate flux splitting algorithm. All results presented in this paper have been obtained with Van Leer's[5] flux vector splitting or with its modified version proposed by Hänel et al.[6] A flux-difference splitting approach, however, could also be considered (e.g., using Roe's algorithm[7]). The new finite-volume scheme using extrapolation formulas similar to Eq. (12) to discretize the advective flux derivatives is named RH3. However, that upwind-biased scheme is not monotone and, therefore, can not be used inside captured shock waves and insufficiently refined shear layers. To overcome that difficulty, a modified form of Eq. (12) using a limiter could be used. It would introduce a progressive transition between RH3 and the first-order upwind scheme RH1. However, we use a different approach. We developed a detector that proved to be extremely sensitive and capable of efficiently identifying regions with an insufficiently refined grid as well as those where wiggles appear for any reason. That detector, which requires the local evaluation of all second-order derivatives of static pressure or/and of flow velocity using centered operators L_i, is presented in the Appendix. At each nodal point, a parameter σ, which is only allowed to be 0 or 1, is then computed from the value of those derivatives. It is evaluated at the beginning of each time step, but kept constant during the different Runge–Kutta substeps. The

space discretization of the time-dependent Navier–Stokes equations then becomes

$$\partial_t s_{m,j} = -[(1-\sigma)\|\mathrm{RH3}\| + \sigma\|\mathrm{RH1}\|]_{m,j} - (\|S_d\|)_{m,j} \qquad (13)$$

where the last term and the term between the brackets are the discretized form of the diffusive and the advective operators, respectively. When σ is equal to 1 at some point, the detector is said to be activated and the RH1 scheme is used at that point. Thanks to the high sensitivity of our detector, it was always found to be activated at all mesh points located inside the numerical shock structure. The sudden transition from one scheme to another was found to be located close to the shock but outside it. For that reason, the local loss of conservativity occurring at that point did not have any practical consequences. We obtained sharp and smooth shocks with the correct strength and location.

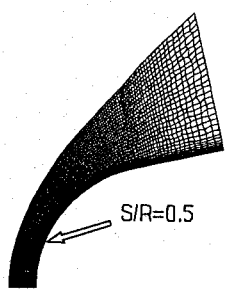
However, we should emphasize an important drawback of RH3: according to property 2 discussed in Sec. III, it is first-order accurate only on irregular meshes and, therefore, can lead to unacceptable diffusive or antidiffusive effects. To overcome that difficulty, a modified version of RH3, named RH4, is actually used. It is obtained by substracting from $\|\mathrm{RH3}\|$ a source/sink-like term that restores the unconditional second-order accuracy according to property 2. Therefore, all our calculations have been performed with a modified form of Eq. (13) obtained by replacing $\|\mathrm{RH3}\|$ by $\|\mathrm{RH4}\|$. The latter expression is defined as follows:

$$\|\mathrm{RH4}\| = \|\mathrm{RH3}\| - [\|E_{1,m,j}\| + \|E_{2,m,j}\|] \qquad (14)$$

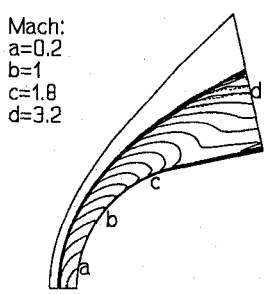
where terms between the brackets are approximate forms of expressions (10) obtained by using the corrected centered discretization operators. Because of the source/sink term, RH4 is no longer strictly conservative. However, it can be considered as almost conservative in the sense that this term is very small in smooth flow regions, that is, in fact, of second-order with respect to the local characteristic mesh sizes. However, the latter remark is not valid inside captured shock waves. This is unimportant because neither RH4 nor RH3 is actually used inside shocks. It is interesting to remark that if we used a cell-centered (instead of a node-centered) mesh, expressions (10) would vanish for stretched grids with rectangular or parallelogram cells. RH4 would then be identical to RH3 and, therefore, conservative.

In our inviscid flow calculations, local values of σ were obtained from a detector using the derivatives of static pressure. In our Navier–Stokes code, a second detector based on the derivatives of flow velocity was also used. Note that the comparison of the values of σ obtained with the two detectors provides an interesting opportunity to selectively identify shocks and insufficiently refined shear layers. In our calculations, σ was automatically chosen equal to 1 at all points where any of the two detectors was found to be activated.

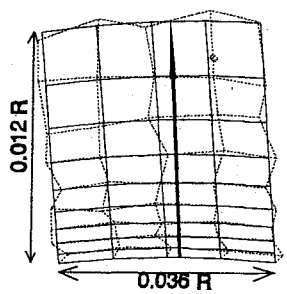
In several preliminary tests, we noticed that the convergence history diagram systematically presented severe oscillations. Quite frequently the solution did not fully converge and eventually became unstable. That behavior can be explained as follows. When for any reason wiggles appear in some parts of the flow, the detector is activated, and the RH1 scheme is used. Because of its important



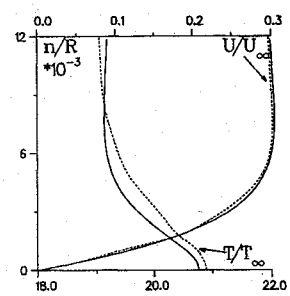
a) 140 × 36 distorted mesh



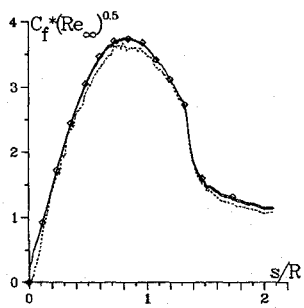
b) Iso-Mach lines, $Pr = 1$



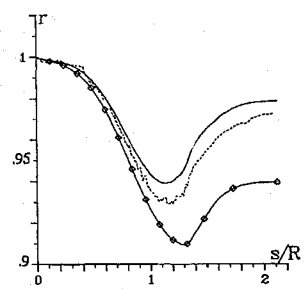
c) Typical portion of regular and distorted mesh in the boundary layer



d) Velocity and temperature profiles. Cross section $s/R = .5$, $Pr = 1$

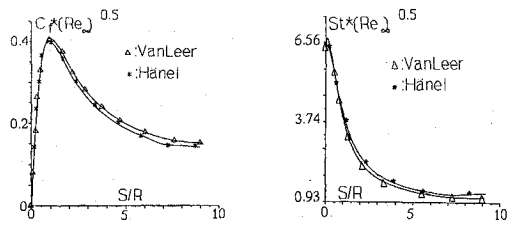


e) Skin friction coefficient



f) Recovery factor

Fig. 3 Sphere-cone configuration semivertex angle 12.5 deg, $M_\infty = 10$; $Re_{\infty,R} = 4.10^5$, $T_\infty = 200$ K; zero angle of attack, adiabatic wall. ——: regular mesh, $Pr = 1$; - - - - - -: distorted mesh, $Pr = 1$; —◇— regular mesh, $Pr = 0.72$, S/R: curvilinear coordinate along body (from the axis), R: sphere radius.



a) Skin friction b) Stanton number

Fig. 4 Isothermal elliptic cylinder, zero angle of attack, M_∞ = 8.15; $R_{\infty,R}$ = 6.26 10^5, T_∞ = 56 °K, T_w = 288 °K, 129 × 49 mesh. s: curvilinear coordinate along body from the nose (intersection with larger principal axis). Principal axes lengths ratio: 0.25, R: nose radius of curvature.

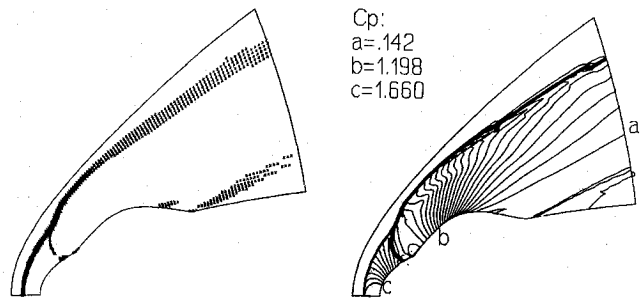
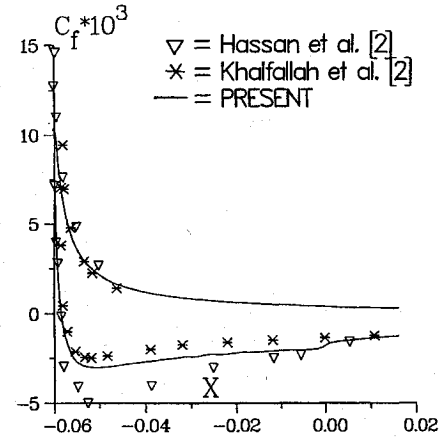


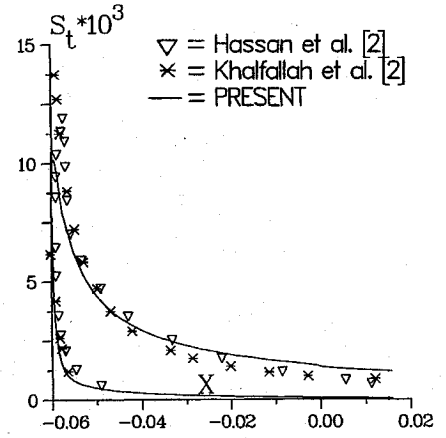
Fig. 5 Isothermal elliptic cylinder with a bump; for flow conditions see Fig. 4.

dissipative character, the RH1 scheme can damp the wiggles out. Outside of the shocks waves, the solution can then appear to be smooth enough, and the RH4 scheme is again used. The oscillations then frequently appear again and convergence can usually not be achieved. That situation frequently occurs in insufficiently refined boundary layers. The following procedure proved to be very efficient to force convergence. Therefore, it was definitely introduced in the code and used in all our calculations. In that procedure, the whole calculation is considered as a sequence of time cycles, each corresponding to a preassigned number of successive time steps (typically of the order of 50–80). A counter is assigned to each node. All counters are initially equal to 0 and reset to 0 at the beginning of each new cycle. At the end of each time step, the number displayed by any counter is systematically increased by 1 when the value of σ computed at the corresponding node is found to have switched from 0 to 1 or conversely since the previous time step. All counters are checked at the end of each cycle. All nodes whose counter indicates a number of switches exceeding a preassigned value are subsequently and definitely forced to use the RH1 scheme until the end of the calculation. The allowed number of switches is chosen equal to a certain percentage of the total number of steps in a cycle (typically of the order of 15–20%). A certain grace period is usually necessary at the beginning of the calculation to allow short waves traveling in the domain to be sufficiently damped. Its duration depends on the initial conditions, but it is usually chosen of the order of 2–4 cycles.

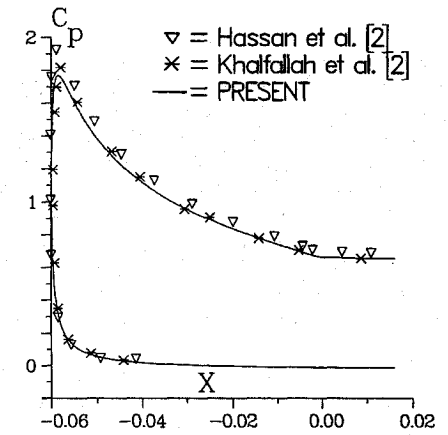
The RH1 scheme is well known to be oversmoothing, to produce too thick boundary layers and, therefore, to significantly underestimate the skin friction and the kinetic heat transfer. It can also become zeroth-order accurate in regions where the grid is severely stretched. For that reason, we never accept a converged solution where the RH1 scheme appears to have been used at a significant number of boundary-layer nodes. When, at a relatively advanced stage of the calculation, that situation seems to occur systematically, the message is understood as follows: "please, refine the grid in this region before going on!" All results presented here have been obtained with grids that were found to be sufficiently fine from that viewpoint. They have eventually been obtained after a successive mesh refinement process which could be made automatic.



a) Skin friction



b) Stanton number

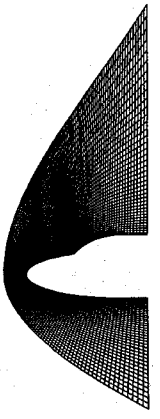


c) Pressure coefficient

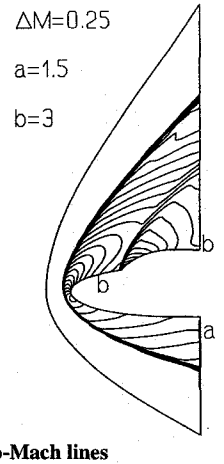
Fig. 6 Isothermal elliptic cylinder at 30-deg angle of attack, M_∞ = 8.15; $R_{\infty,\text{meter}}$ = 1.67 10^7, T_∞ = 56 K, T_w = 288 K, 323 × 80 mesh. Half length of larger principal axis: 0.06 meter. Principal axes lengths ratio: 0.25.

VI. Runge–Kutta Scheme and Boundary Conditions

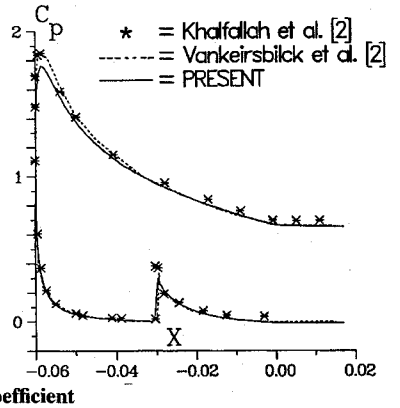
A pseudotime-dependent approach with a variable time stepping is used to compute updated values of the conservative variables at time level $(l + 1)$ from their known values at time level l. Time integration of Eq. (13) is performed with a multistep Runge–Kutta scheme. We essentially considered some three- and four-step schemes that are only first-order accurate in time and introduced a purely transient additional dissipative effect which does not influence the steady solution. The scheme used was selected as the result of a tradeoff between the maximum allowable time step and the strength of transient dissipative effects. Boundary conditions described subsequently are used at the end of each predictor and corrector step.



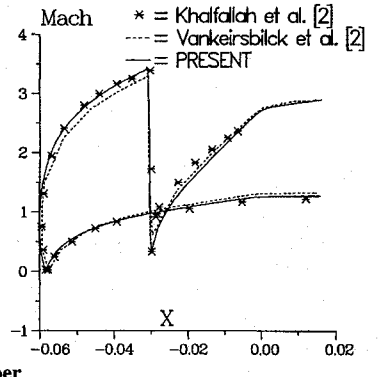
a) 175 × 40 mesh



b) Iso-Mach lines



c) Pressure coefficient



d) Mach number

Fig. 7 Double-ellipse test case,[2] inviscid flow $M_\infty = 8.15$, angle of attack 30 deg.

All conservative and primitive flow variables are obviously fixed on the supersonic inflow boundary. At each node of the supersonic outflow boundary, the updated values of all conservative flow variables are obtained from the Runge–Kutta scheme applied to the discretized form [Eq. (13)] of all Navier–Stokes equations using some dummy control points located downstream of the boundary. To understand the treatment of solid wall boundaries, it is essential to notice that the updated values of all flow variables at time level $(l + 1)$ are first computed at all nodal points located inside the domain before updating the values of temperature and static pressure at solid wall nodal points. For nonisothermal walls, the boundary condition on temperature includes a normal derivative of the latter. It is considered as a combination of x and y derivatives, which are discretized at the boundary nodes using the corrected operators described in Sec. II. The discretized boundary condition then couples the values of the temperature at nine different nodes, all written at the same time level $(l + 1)$. As the temperature has already been updated at six of these points, that boundary condition appears as an algebraic equation coupling the unknown temperatures at three successive boundary nodes, thus leading to a tridiagonal system, whose solution provides the updated temperature at all points of the wall. Accounting for the no-slip boundary conditions and again using the corrected operators, we then compute the updated values of the diffusive flux vectors at all mesh points including those located on the boundary. By suitably combining the x and y momentum equations written at the solid wall and using the equation of state, the following schematic expression of the normal derivative of pressure is obtained:

$$\partial_n p = 1/k \cdot [\text{diffusive derivatives}] \cdot p \qquad (15)$$

where k is proportional to the temperature, and the term between the brackets is a combination of the first-order derivatives of the updated diffusive flux vectors. These derivatives are computed using the corrected operators. As a result, the pressure p at the wall nodal point is the only unknown remaining in the right-hand side of Eq. (15). Using for the left-hand side term the same discretization as that used for the normal derivative of temperature, a tridiagonal system is obtained that is solved to compute the static pressure.

VII. Results

In this section, we use the following definitions of the pressure coefficient C_p, the skin friction coefficient C_f, the recovery factor r, and the Stanton number St:

$$C_p = (p - p_\infty)/\left(0.5\rho_\infty U_\infty^2\right); \qquad C_f = \tau_w/\left(0.5\rho_\infty U_\infty^2\right)$$
$$r = (T_w - T_\infty)/(T_{t\infty} - T_\infty); \qquad (16)$$
$$St = q_w/[\rho_\infty U_\infty c_p (T_{t\infty} - T_w)]$$

where $T_{t\infty}$ is the total temperature of the flow, c_p is the specific heat of the gas at constant pressure, and τ_w, T_w, q_w denote the shear stress, the static temperature, and the heat flux distribution along the body surface, respectively. U_∞, p_∞, ρ_∞, and T_∞ are the velocity, the static pressure, the specific mass, and the static temperature of the incoming flow, respectively. The fluid is a perfect diatomic gas with a specific heat ratio of 1.4. If not stated otherwise, the Prandtl number is 0.72 and the advective fluxes are computed with Van Leer's flux vector splitting.[5] The grid is severely stretched near the wall. For all blunt body flow calculations, boundary-layer thickness is typically covered by 8–12 mesh cells.

Figure 3 presents results obtained for the axisymmetric flow with a Mach number of 10 over an adiabatic sphere-cone configuration with two different Prandtl numbers, 0.72 and 1. Two 140×36 meshes are used for these calculations: a relatively regular grid and an irregular mesh obtained by applying a random grid distortion procedure to the latter. The distorted grid is shown in Fig. 3a. Velocity and temperature profiles obtained in a typical cross section of the boundary layer are presented in Fig. 3d. An enlarged view of the two grids used near the wall in that region is shown in Fig. 3c. Note that for the sake

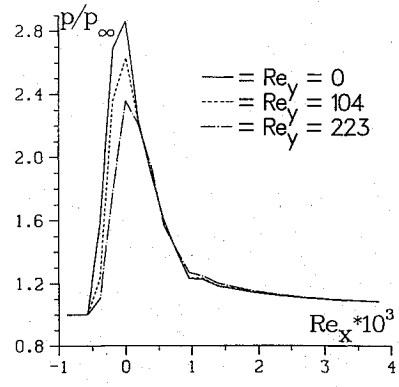
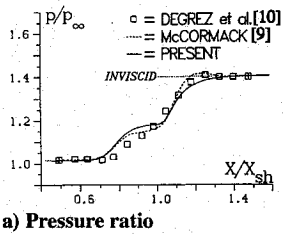
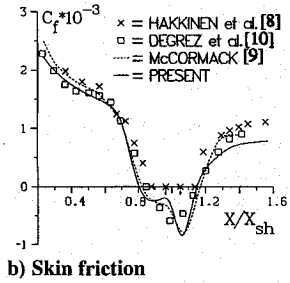


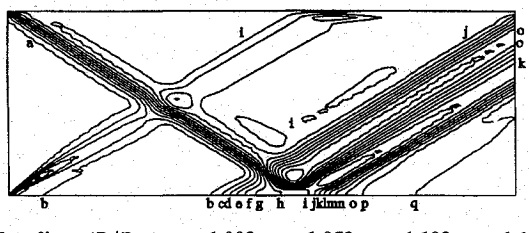
Fig. 8 Leading-edge interaction for adiabatic flat plate, local static to upstream pressure ratio along and ahead of the plate; $M_\infty = 2$; $R_{\infty,\text{meter}} = 6.04\ 10^6$, $T_\infty = 156$ K.



a) Pressure ratio

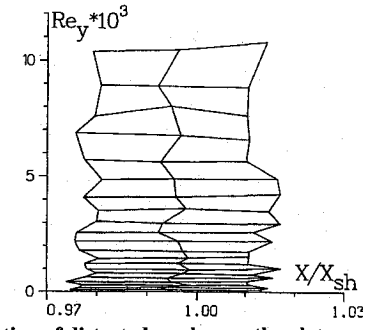


b) Skin friction

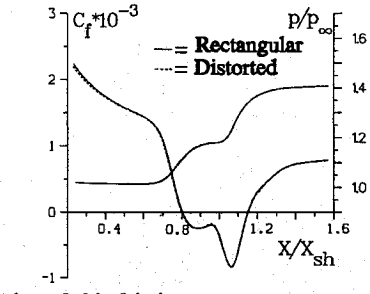


c) Isobar lines (P/P_∞), $a = 1.002$; $c = 1.052$; $e = 1.102$; $g = 1.152$; $i = 1.202$; $k = 1.252$; $m = 1.302$; $o = 1.352$; $q = 1.402$

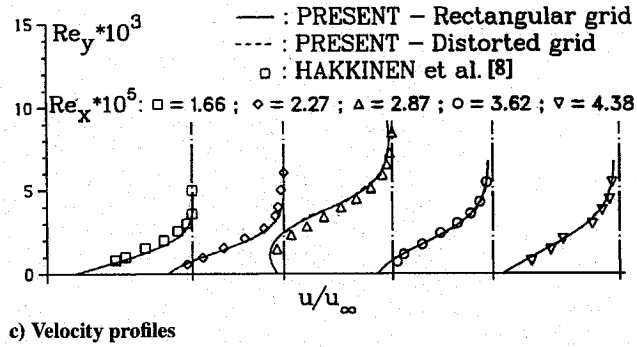
Fig. 9 Two-dimensional shock/boundary-layer interaction on adiabatic flat plate, $M_\infty = 2$; $R_{\infty,xs} = 2.96\ 10^5$, $T_\infty = 156$ K, incident shock angle 32.6 deg, 151 × 52 mesh. Inflow boundary located at distance: 0.05 x_{sh} upstream of leading edge. x_{sh}: distance between leading edge and incident shock impingement point (from inviscid theory).



a) Typical portion of distorted mesh near the plate



b) Pressure ratio and skin friction



c) Velocity profiles

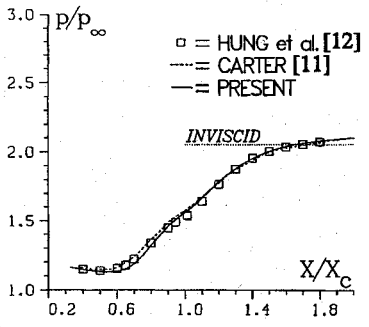
Fig. 10 Two-dimensional shock/boundary-layer interaction, comparison of results obtained with a rectangular and a severely distorted mesh; same flow characteristics as in Fig. 9.

of clarity, two different scales have been used in that figure for the coordinates parallel and perpendicular to the wall. As a result, the mesh line distortions are much more severe than they look. Results obtained with the two grids are found to be in excellent agreement with each other. The differences in the skin friction coefficient and in the recovery factor computed along the wall does not exceed 1–2% (Figs. 3e and 3f). It is interesting to note that for $Pr = 1$, the use of the classical boundary-layer approximation would lead to a constant value of $r = 1$ and would, therefore, overestimate the wall recovery temperature.

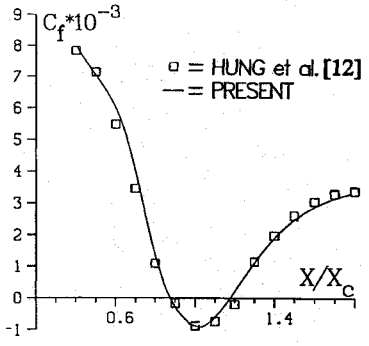
Comparative inviscid flow calculations over the sphere-cone configuration at a Mach number of 10 have been performed with our new code and with an Euler finite-volume code using a classical second-order upwinding with a limiter. For the regular grid, both codes lead to very similar results (not presented here). For the distorted mesh, however, no convergence could be achieved with the second-order upwind scheme, whereas results obtained with our code remained almost unchanged. Comparative results could not be obtained for viscous flows because all of our attempts to solve the Navier–Stokes equations with the second-order upwind scheme failed; convergence could not be achieved even on moderately stretched unskewed meshes.

Our code has also been used to compute the viscous flow at a Mach number of 8.15 over an isothermal elliptic cylinder at zero incidence. Figure 4 provides an interesting comparison of skin friction coefficient and Stanton number obtained using the splitting algorithms of Van Leer[5] and Hanel et al.[6] From this calculation (and from others on similar bodies), we conclude that the two algorithms do not lead to significant differences in the results. A bump simulating a canopy has then been added to the elliptic cylinder. Some of the results obtained for this configuration with a 129 × 49 grid are presented in Fig. 5. The results demonstrate the quality of shock waves capture, as well as the sensitivity and reliability of our detector.

The isothermal elliptic cylinder already mentioned has been placed at 30-deg angle of attack. In Fig. 6, $X = 0$ corresponds to the two points located at the intersection of the ellipse with its smaller principal axis. Downstream of these points, the profile is no longer elliptic but is made of the two tangent planes parallel to the chord. The Mach number is the same as for the calculation at zero incidence, but the Reynolds number is different. The flow characteristics are the same as those used in a recent workshop organized in the framework of the Hermes Research and Development Program.[2] Figure 6 shows the computed skin friction coefficient, the Stanton number, and the pressure coefficient. Our results are compared to those obtained by two studies in Ref. 2 (Khalfallah et al. and Hassan et al.). For the pressure coefficient and the Stanton number, the agreement is generally good except near the nose. For the skin friction, the agreement with the results of Khalfallah et al. found in Ref. 2 is also relatively good. The dispersion in the results obtained by several other participants to the workshop is very large. These results are not presented here.



a) Pressure ratio



b) Skin friction

Fig. 11 Flow over an adiabatic compression ramp, $M_\infty = 3$; $Re_{\infty,xc} = 1.68\ 10^4$, $T_\infty = 216.7$ K, ramp angle 10 deg, 175 × 60 mesh.

Typical results obtained with the Euler version of our code are shown in Fig. 7. We consider the inviscid flow over at 30-deg angle of attack in a flow with a Mach number of 8.15. This is also a test case proposed in aforementioned workshop. The 175 × 40 mesh we used and the computed iso-Mach lines are shown in Figs. 7a and 7b. Figures 7c and 7d give the pressure coefficient and the Mach number along windward and leeward sides. They are compared to the results of Khalfallah et al. and of Vankeirsbilck and Deconinck, both found in Ref. 2.

We performed a Navier–Stokes calculation of a laminar boundary layer developing along a thin adiabatic flat plate at zero incidence in a uniform flow with a Mach number of 2. The inflow boundary is located upstream of the leading edge. A rectangular grid is used for that calculation. It is very fine and severely stretched in the axial direction near the leading edge. The aim is to capture the leading-edge interaction effect. The latter appears clearly in Fig. 8, which gives the pressure distribution along the plate and upstream of it, as well as in different sections parallel to the plate. The far-field boundary conditions were then modified to produce an incident oblique shock with an inclination angle of 32.6 deg. Two different 151 × 52 grids were used to solve that shock/boundary-layer interaction problem: a stretched rectangular mesh and a mesh with severe skewness distortions generated from the latter. Figures 9a and 9b show the pressure and the skin friction computed along the plate with the rectangular grid. They are compared to the experimental results of Hakkinen et al.[8] and to the numerical results obtained by MacCormack[9] and by Degrez et al.[10] The agreement is generally good. The pressure plateau we obtain also appears in the results of MacCormack,[9] but not in those of Degrez et al.[10] That plateau is also present in the experimental results of Hakkinen et al.[8] (not shown). The compression and expansion waves generated by the separation bubble and the waves due to the leading-edge interaction appear clearly in Fig. 9c. A typical portion of the distorted mesh near the plate is shown in Fig. 10a. Figure 10b provides an interesting comparison of the pressure and the skin friction obtained with the rectangular and the distorted grids, respectively. They almost coincide. The same observation also applies to the velocity profiles shown in Fig. 10c. The agreement with the profiles measured by Hakkinen et al. is good. These figures again demonstrate the remarkably weak sensitivity of our code to mesh distortions.

Figure 11 shows results obtained for a flow with a Mach number of 3 over a 10-deg angle compression corner. The inflow boundary is again located upstream of the plate, and the 175 × 60 mesh is designed to capture the leading-edge interaction effect. Good agreement is found with the numerical results of Carter[11] and of Hung and MacCormack.[12]

VIII. Concluding Remarks

A new upwind-biased finite-volume scheme for the solution of the compressible Navier–Stokes equations has been presented. It is designed to unconditionally ensure the consistency of the discretized diffusive flux vector derivatives and the second-order accuracy in the approximation of the advective derivatives on severely distorted meshes. A new very sensitive detector has been developed which automatically and selectively identifies shocks and boundary layers discretized with an insufficiently refined grid. It is used to switch to a classical first-order upwind scheme in those regions to avoid oscillations. A code using the new scheme has been successfully tested on different plane and axisymmetric flows including hypersonic blunt body flows, a supersonic flow over a compression ramp, and a shock/boundary-layer interaction on a flat plate. Results generally show good agreement with those obtained with other codes, as well as with experimental results for the shock/boundary-layer interaction. Calculations performed on the same problems using a regular grid and a grid with severe distortions in the whole domain including boundary layers demonstrate the remarkably weak sensitivity of our code to mesh stretching and skewness.

The new scheme presented in this paper can be generalized to unstructured grids. A code based on this generalized scheme using cell-centered unstructured meshes with any polygonal cells is presently under development. In the future, we plan to use it to solve the Navier–Stokes equations with adaptive patched grid refinements.

In this paper, results obtained with randomly and artificially distorted meshes have been presented to demonstrate the capabilities of our scheme. However, the question arises as to whether it is really superior to more traditional schemes for meshes we actually encounter in practice. For structured grids around relatively simple two-dimensional configurations, its use could probably not be justified because simpler and more economical schemes would presumably lead to very similar results, at least for inviscid flows. Our main goal was to provide some evidence that our scheme would certainly be quite appropriate for irregular meshes that are systematically obtained when performing multiblock adaptive patched grid refinements and for completely unstructured meshes with triangular elements. Additionally we believe that a generalized version of our code would be very useful for three-dimensional flow calculations because the generation on very regular meshes around complex three-dimensional configurations is a very difficult and time-consuming task.

Appendix: Detector

First consider the static pressure field p defined on an uniform square grid with a constant mesh size h. The second-order derivative of p in a direction forming the angle θ with the x axis is given by

$$\partial^2_{ss} p = (\cos^2\theta)\partial^2_{xx} p + \sin^2\theta\,\partial^2_{yy} p + 2\sin\theta\cos\theta\,\partial_{xy} p \qquad \text{(A1)}$$

At any mesh point (m, j), the maximum value $DD2\max$ of $|\partial^2_{ss} p|$ in all directions can be computed from the following formula:

$$(DD2\max)_{m,j} = |0.5(a+b)| + [c^2 + 0.25(a-b)^2]^{0.5} \qquad \text{(A2)}$$

where

$$a = \left(\partial^2_{xx} p\right)_{m,j}; \qquad b = \left(\partial^2_{yy} p\right)_{m,j}; \qquad c = \left(\partial^2_{xy} p\right) \qquad \text{(A3)}$$

A sensor s that can be used as a numerical detector is then calculated at each nodal point

$$s_{m,j} = h^2/\Delta P (DD2\max)_{m,j} \qquad \text{(A4)}$$

where ΔP is a typical characteristic value of $|p - p_\infty|$ in the flow, which can be evaluated as explained subsequently. Note that the

following relation can be used to define a local characteristic length of the pressure field:

$$l^2_{m,j} = \Delta P/(DD2\max)_{m,j}$$

If it corresponds to an important local physical characteristic length and if the mesh is fine enough in that region, s is very small, i.e., of the order of $(h/l)^2$. However, it can be of the order of 1 or even larger if the mesh is too coarse. The same observation also applies if severe numerical oscillations appear or if the mesh point is located inside a shock wave. To make our detector even more sensitive, we use a slightly different formula to compute s

$$s_{m,j} = \left[\frac{hL_p}{\Delta P}(DD2\max)_{m,j}\right]^2 \tag{A5}$$

where L_p is the smallest geometrical (or physical) characteristic length that should be correctly discretized. It can easily be evaluated a priori and was taken equal to the nose radius of curvature in all our blunt body flow calculations. From the definition of L_p, it is clear that l is never found very small with respect to L_p in correctly discretized smooth flow regions. As a result, s is again very small, i.e., of the order of $(h/l)^2(L_p/l)^2$ in those regions. It is, however, of the order of $(L_p/h)^2$ and, therefore, very large inside shock waves and in regions with spurious oscillations.

Formula (A5) should be generalized for nonuniform structured meshes with quadrilateral elements. For that purpose, we define $2h_{1,mj}$ as the distance between the mesh points $(m-1, j)$ and $(m+1, j)$, and we denote by $(\partial^2_{11}p)_{mj}$ the second-order derivative in the direction parallel to the straight line joining those two points. It can easily be computed from a formula similar to Eq. (A1). In that formula, the second-order derivatives with respect to x and y are evaluated using the space-centered operators defined in Sec. II. Similarly, $2h_{2,m,j}$ and $(\partial^2_{22}p)_{m,j}$ denote the distance between nodal points $(m, j-1)$ and $(m, j+1)$ and the second-order derivative in a direction parallel to the line joining those points, respectively. The generalized formula defining the detector is given by

$$s^{(1)}_{m,j} = \left[\frac{Lp}{\Delta P}(DD2\max)_{m,j}\right]^2 \tag{A6}$$

where $DD2\max$ is still computed from Eq. (A2), but with values of a, b, and c now defined as follows:

$$a = h_{1,m,j}\left(\partial^2_{11}p\right)_{m,j}; \qquad b = h_{2,m,j}\left(\partial^2_{22}p\right)_{m,j};$$
$$c = (h_{1,m,j}h_{2,m,j})^{0.5}\left(\partial^2_{12}p\right)_{m,j} \tag{A7}$$

If $s^{(1)}_{m,j}$ is smaller than a user-defined positive constant ε, the pressure field is assumed to be smooth and the mesh is considered as being sufficiently fine in that region. If it is larger, the detector has to be activated, and the RH1 scheme has to be used at the point (m, j). For all our blunt body flow calculations, ε was chosen of the order of 0.1.

If the mesh is not fine enough at the point (m, j), the centered operators $L_i^{(m,j)}$, which have been constructed from local Taylor series expansions, are very inaccurate and extrapolation formula (12) does not provide a reliable estimate of the fluxes on control volume sides. As a result, the detector should also be activated and the RH1 scheme should to be used at the neighboring mesh points $(m+k, j)$ and $(m, j+k)$ corresponding to $k = -1$ and $+1$. For that reason, the final value of the detector is computed as follows:

$$s^{(2)}_{m,j} = \max\left[s^{(1)}_{m+k,j}, s^{(1)}_{m,j+k}\right] \qquad \text{for all } k = -1, 0, +1 \tag{A8}$$

We then evaluate

$$(\sigma_P)_{m,j} = 0 \quad \text{if} \quad s^{(2)}_{m,j} \le \varepsilon, \qquad (\sigma_P)_{m,j} = 1 \quad \text{otherwise} \tag{A9}$$

For subsonic and transonic flow calculations, a constant value of ΔP can be used and estimated a priori because the strength of eventual shock waves is moderate. For flows around airfoils, for example, it can be obtained from a rough estimate of the maximum and minimum pressure coefficients on the surface. For hypersonic flows with bow shocks, the shock is still relatively strong at a relatively large distance from the axis but is much weaker than near the latter. As a result, if a constant value of ΔP was used, important parts of the bow shock could not be detected. For that reason, the following local values of ΔP have been used in our supersonic/hypersonic blunt body flow calculations

$$(\Delta P)_{m,j} = \max[|p - p_\infty|_{m+k,j+r}] \qquad \text{for all } k, \quad r = -1, 0, +1$$

They are computed at the beginning of each new time step.

For inviscid flows, the local values of σ used in formula (13) is taken equal to $(\sigma_p)_{m,j}$. For viscous flows, another sensor σ_q also has to be computed to detect insufficiently refined boundary layers. It is obtained from formulas completely similar to those used to evaluate σ_p. However, second-order derivatives of the local flow velocity $q = (u^2 + v^2)^{0.5}$ and a characteristic length L_q different from L_p are used in formulas (A6) and (A7). L_q is an a priori estimate of boundary-layer thickness obtained from

$$L_q = L_p(Re_{\text{eff}})^{-0.5}$$

where Re_{eff} is an effective Reynolds number computed from an effective viscosity μ_{eff}. The latter is the viscosity corresponding to an estimated wall temperature. In our adiabatic flow calculations, that temperature was taken equal to the total temperature of the incoming flow. A local characteristic velocity $(\Delta Q)_{m,j}$ should also be used instead of ΔP in formula (A6). It cannot be chosen equal to the local flow velocity $q_{m,j}$ because the latter is very small near the wall. Therefore, it is evaluated as follows:

$$(\Delta Q)_{m,j} = \max(q_{m,j}, w_{m,j})$$

where $w_{m,j}$ is a fictitious velocity. To calculate it, an estimate $\bar{p}_t$ of the constant total pressure that would be obtained along the wall if the flow was inviscid is made a priori. It is taken equal to the estimated stagnation pressure. The fictitious velocity is then evaluated as a fraction of the local static pressure and temperature $p_{m,j}$ and $T_{m,j}$

$$1 + \frac{\gamma - 1}{2}\left(\frac{w_{m,j}}{a_{m,j}}\right)^2 = \left(\frac{\bar{p}_t}{p_{m,j}}\right)^{\frac{\gamma-1}{\gamma}}$$

where $a_{m,j}$ is the local sound speed corresponding to the temperature $T_{m,j}$, and γ is the specific heat ratio.

Acknowledgments

The work of P. Rogiest was supported by a grant provided by I.B.M. Belgium and awarded by the Belgian National Fund for Scientific Research. Calculations presented in this paper have been performed on the IBM 3090-200E computer of the University of Liège. Part of the computing costs was funded by the Belgian Science Policy Office. The authors gratefully acknowledge these supports.

References

[1]Renard, E., and Essers, J. A., "An Analysis of Severe Grid Distortion Effects on the Accuracy of Some Discretization Schemes for Convection-Diffusion Equations," *Numerical Grid Generation in Computational Fluid Mechanics'88*, pp. 845–853.

[2]Desideri, J. A., Glowinski, R., and Periaux, J. (eds.), *Hypersonic Flows for Reentry Problems*, Vols. 1 and 2, Springer–Verlag, 1991.

[3]Essers, J. A., and Renard, E., "An Implicit Flux-Vector Splitting Finite-Element Technique for an Improved Solution of Compressible Euler Equations on Distorted Grids," *11th International Conference on Numerical Methods in Fluid Dynamics*, edited by Dwoyer, Hussaini and Voigt, Lecture Notes in Physics 323, Springer–Verlag, 1989, pp. 251–255.

[4]Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," AIAA Paper 93-0668, Jan. 1993.

[5]Van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Lecture Notes in Physics 170*, Springer–Verlag, 1982, pp. 507–512.

[6]Hänel, D., Schwane, R., and Seider, G., "On the Accuracy of Upwind Schemes for the Solution of the Navier–Stokes Equations," AIAA Paper 87-1105, 1987.

[7]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–72.

[8]Hakkinen, R. J., Greber, I., Trilling, L., and Abarbanel, S. S., "The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer," NASA Memo 2-18-59W, 1959.

[9]MacCormack, R. W., "An Efficient Numerical Method for Solving the Time-Dependent Compressible Navier–Stokes Equations at High Reynolds Numbers," NASA TMX 129, 1976.

[10]Degrez, G., Boccadoro, C. H., and Wendt, J. F. W., "The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer Revisited. An Experimental and Numerical Study," *Journal of Fluid Mechanics*, Vol. 177, 1987, pp. 247–267.

[11]Carter, J. E., "Numerical Solution of the Navier–Stokes Equations for the Supersonic Laminar Flow over a Two-Dimensional Compression Corner," NASA TR R-385, 1972.

[12]Hung, C. M., and MacCormack, R. W., "Numerical Solution of Supersonic and Hypersonic Laminar Compression Corner Flows," *AIAA Journal*, Vol. 14, No. 4, 1976, pp. 475–481.