AIAA JOURNAL
Vol. 38, No. 11, November 2000

Accurate and Efficient Discretization of Navier–Stokes Equations on Mixed Grids

Andreas Haselbacher* and Jiri Blazek†
ALSTOM Power, Ltd., Segelhof, 5405 Baden-Dättwil, Switzerland

The discretization of Navier–Stokes equations on mixed unstructured grids is discussed. Because mixed grids consist of different cell types, the question arises as to how the discretization should treat these cell types in order to result in a stable and accurate solution method. This issue is addressed in relation to the discretization of inviscid and viscous fluxes. The discretization of the inviscid fluxes is carried out with both a centered and an upwind scheme. For the centered scheme, problems exist on mixed grids with the damping properties of the fourth-difference operator. The problems with the fourth-difference operator are investigated for two stencils by both truncation error and Fourier analysis. It is shown that one form exhibits superior damping for high frequencies, which is corroborated using numerical experiments. For the upwind scheme, a commonly used gradient-reconstruction method based on the Green–Gauss theorem does not show satisfactory behavior on mixed grids. Another gradient reconstruction method based on a Taylor series expansion previously derived in two dimensions is extended to three dimensions. This method is more accurate on mixed grids but requires more storage.

Nomenclature

$\boldsymbol{D}_0$	=	dissipation term at node 0
$\mathcal{E}_{0i}$	=	set of edges incident to node 0
$\boldsymbol{F}_c$	=	inviscid flux vector
$\boldsymbol{F}_v$	=	viscous flux vector
$L(\phi)$	=	pseudo-Laplacian operator applied to ϕ
$\boldsymbol{n}_{0i}$	=	unit normal face vector at edge $0i$
$\boldsymbol{Q}$	=	source vector
$\boldsymbol{r}$	=	position vector
$\boldsymbol{r}_{0i}$	=	unit tangential vector along edge $0i$
S_{0i}	=	area of face associated with edge $0i$
$\boldsymbol{W}$	=	vector of conservative variables
θ_{0i}	=	geometric weights associated with edge $0i$
Λ_{0i}	=	spectral radius averaged over edge $0i$
λ_0	=	limiter function at node 0
ϕ	=	general quantity
ϕ_x	=	phase angle in x direction
ϕ_y	=	phase angle in y direction
Ω_0	=	control volume at node 0
$(\nabla\phi)_0$	=	gradient of ϕ at node 0

Introduction

UNSTRUCTURED grids composed of geometric simplices, i.e., triangles in two dimensions and tetrahedra in three dimensions, have demonstrated the capability of producing high-quality solutions for inviscid flow simulations; see, e.g., Frink et al.[1] The simulation of viscous flows requires the generation of highly stretched triangles and tetrahedra. For simplicial elements, the stretching invariably leads to skewing. The application of simplicial unstructured grids to viscous flows has been far less successful. Because unstructured grids may consist of arbitrary combinations of polyhedra, it seems sensible to use cell shapes in the viscous regions that do not become skewed with stretching, such as hexahedra and prisms. When combined with tetrahedral cells away from viscous regions, this leads to grids composed of several cell types, which are referred to as mixed grids in this paper. Several authors have presented solution methods for mixed grids; see, e.g., Mavriplis and Venkatakrishnan,[2] Khawaja et al.,[3] Coirier and Jorgenson,[4] Haselbacher et al.,[5] and Blazek et al.[6] Mixed grids are sometimes also referred to as hybrid grids, but are not to be confused with combined structured–unstructured grids; see, e.g., Shaw.[7]

The existence of different cell types in the grid raises the question as to how they are treated by the solution method. It seems intuitively appealing not to distinguish between the various cell types. This leads to the fundamental challenge that a discretization method needs to be developed that is stable and accurate for grids consisting uniquely of tetrahedra, prisms, pyramids, and hexahedra, or an arbitrary combination of these. This issue has not yet received enough attention. In Ref. 5, algorithms that do not distinguish between different cell types were referred to as grid transparent.

The authors' interest was aroused because problems were encountered during the development of a numerical solution method for three-dimensional viscous flows in turbomachinery. These problems were directly related to the need for an accurate and efficient discretization of Navier–Stokes equations on mixed grids. The problems occurred because the solution method of an initial version of the code was based on that developed previously for tetrahedral grids and assumed to work on mixed and hexahedral grids also. This was found not to be the case and led to investigations, the results of which are reported here.

The aim of this paper is to draw attention to the problems that arise from the discretization on mixed grids, to describe and analyze them in detail, and to offer some solutions.

Governing Equations

The fluid flows considered in this paper are governed by three-dimensional Navier–Stokes equations. Written in a time-dependent integral form for a control volume Ω with a surface element $\mathrm{d}S$, the equations read

$$\frac{\partial}{\partial t}\int_\Omega \boldsymbol{W}\,\mathrm{d}\Omega + \oint_{\partial\Omega}(\boldsymbol{F}_c - \boldsymbol{F}_v)\,\mathrm{d}S = \int_\Omega \boldsymbol{Q}\,\mathrm{d}\Omega \qquad (1)$$

where the various quantities are defined in Ref. 6.

Solution Algorithm

The discretization of the governing equations follows the method of lines, which decouples the spatial and the temporal discretizations. The spatial discretization is carried out with the finite-volume method. The conservative variables are stored at the grid nodes. The control volumes are formed from the median dual of the grid. These can be defined for arbitrary cell types as the polyhedra constructed from connecting cell and face centroids to the edge midpoints, as

Received 10 June 1999; revision received 10 March 2000; accepted for publication 20 March 2000.

*Research Scientist, Section CHAAT.T2; currently Research Scientist, Center for Simulation of Advanced Rockets, University of Illinois at Urbana–Champaign, 3244 Digital Computer Laboratory MC-258, 1304 West Springfield Avenue, Urbana, IL 61801; haselbac@uiuc.edu.

†Project Manager, Section CHAAT.T2; currently Research Scientist, Aerodynamics of Turbomachinery; jiri.blazek@power.alstom.com.

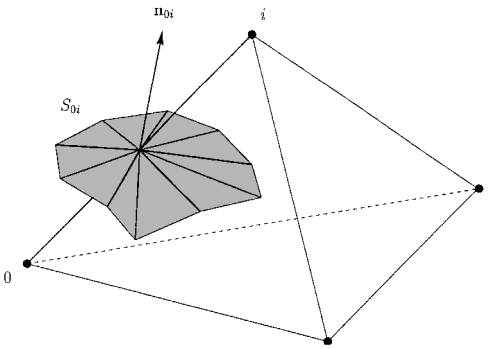


Fig. 1 Edge 0*i* linking nodes 0 and *i* with the associated control-volume face.

shown in Fig. 1. The surface integral of the convective and the viscous fluxes are approximated by a sum over the faces of each control volume.

Assuming the source term to be constant inside the control volume, the discretized form of Eq. (1) at node 0 can be written as

$$\Omega_0 \frac{\mathrm{d}\boldsymbol{W}_0}{\mathrm{d}t} = -\sum_{0i \in \mathcal{E}_0} (\boldsymbol{F}_c - \boldsymbol{F}_v)_{0i} S_{0i} + \boldsymbol{Q}_0 \Omega_0 \tag{2}$$

where S_{0i} is the control-volume face with unit normal $\boldsymbol{n}_{0i}$ separating nodes 0 and i and $\mathcal{E}_{0i}$ denotes the set of edges incident to node 0. Equation (2) represents a system of ordinary differential equations, which has to be integrated in time to reach a steady-state solution.

In what follows, the discretization is described for the control volume around node 0. With the edge-based data structure,[8] the computation of inviscid and viscous fluxes in the interior of the solution domain can be carried out by looping over edges.

Discretization of Convective Fluxes

Two methods were considered for the computation of the inviscid fluxes.

The first method is the central-difference scheme of Jameson, Schmidt, and Turkel,[9] originally derived on structured grids and extended by Jameson et al.[10] to tetrahedral grids. This method will be referred to as the JST scheme in what follows. Because of its low demands on storage and processing time, the JST scheme appears attractive for flow simulations in the industrial environment.

The second method is the upwind scheme of Roe.[11] Compared with the JSTscheme, it shows much improved resolution of shear layers and sharp gradients at the expense of significantly higher computational cost.

Central Scheme

The JST scheme stabilizes the central differences by adding a dissipation term $\boldsymbol{D}_0$ such that Eq. (2) becomes

$$\Omega_0 \frac{\mathrm{d}\boldsymbol{W}_0}{\mathrm{d}t} = -\sum_{0i \in \mathcal{E}_0} (\boldsymbol{F}_c - \boldsymbol{F}_v)_{0i} S_{0i} + \boldsymbol{D}_0 + \boldsymbol{Q}_0 \Omega_0 \tag{3}$$

The dissipation term consists of a blend of second- and fourth-order differences.[9] The pressure switch, which is used as the blending device, is of the standard form.[10] The construction of the second- and the fourth-order differences in two dimensions is discussed below. Two methods were investigated for the fourth-difference operator.

Second-Difference Operator

The second-order differences are represented by pseudo-Laplacian operators. The pseudo-Laplacian may be thought of as an area-weighted Laplacian operator. The pseudo-Laplacian used in this paper is that of Holmes and Connell,[12] which has the property that it is identically zero for linear functions on arbitrary grids.

Applied to a quantity ϕ, the pseudo-Laplacian $L(\phi)$ takes the form

$$L(\phi)_0 = \sum_{0i \in \mathcal{E}_0} \theta_{0i} (\phi_i - \phi_0) \tag{4}$$

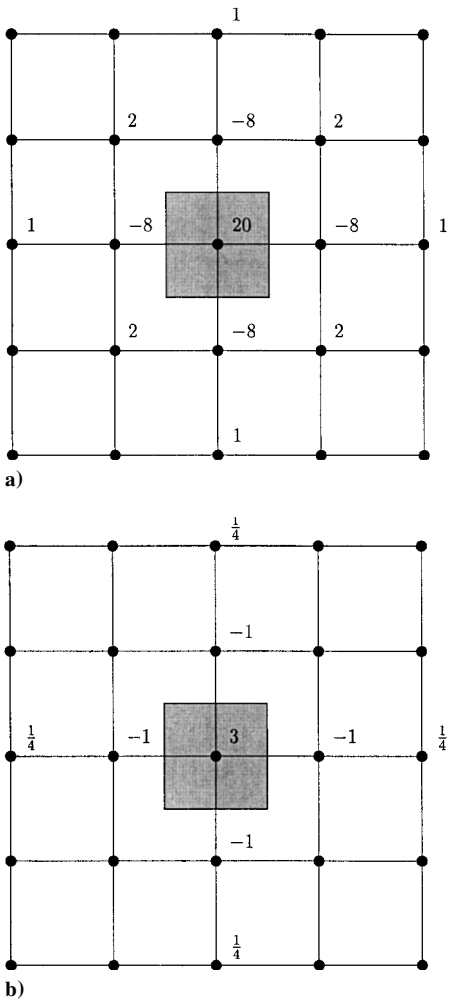


Fig. 2 Comparison of stencil weights obtained on quadrilateral grids with a) first method and b) second method for obtaining fourth differences.

where the definition of the geometric weights θ_{0i} can be found in Ref. 13.

Fourth-Difference Operator, Method 1

The first method evaluates the fourth-difference operator as the pseudo-Laplacian of the pseudo-Laplacian, as originally suggested by Jameson et al.[10] on tetrahedral grids. This gives rise to the biharmonic operator

$$L[L(\phi)_0]_0 \approx \frac{\partial^4 \phi}{\partial x^4} + 2\frac{\partial^4 \phi}{\partial x^2 \partial y^2} + \frac{\partial^4 \phi}{\partial y^4} \tag{5}$$

The stencil that arises from this construction on a uniform quadrilateral grid with grid-spacing unity is shown in Fig. 2a. A Taylor series analysis shows that the first method gives rise to a truncation error of

$$\mathrm{TE} = \frac{1}{6}\left(\frac{\partial^6 \phi}{\partial x^6} + \frac{\partial^6 \phi}{\partial x^4 \partial y^2} + \frac{\partial^6 \phi}{\partial x^2 \partial y^4} + \frac{\partial^6 \phi}{\partial y^6} \right) + \mathrm{HOT}$$

where HOT stands for higher-order terms. The final form of the artificial dissipation term for the first method is for node 0:

$$\boldsymbol{D}_0 = \sum_{0i \in \mathcal{E}_0} \Lambda_{0i} \varepsilon_{0i}^{(2)} \theta_{0i} (\boldsymbol{W}_i - \boldsymbol{W}_0) - \sum_{0i \in \mathcal{E}_0} \Lambda_{0i} \varepsilon_{0i}^{(4)} \theta_{0i} [L(\boldsymbol{W})_i - L(\boldsymbol{W})_0] \tag{6}$$

In Eq. (6), $\varepsilon_{0i}^{(2)}$ and $\varepsilon_{0i}^{(4)}$ are the coefficients defined in Ref. 10 and Λ_{0i} is the edge average of the spectral radius of the Jacobian of the inviscid fluxes.

Fourth-Difference Operator, Method 2

The second method is based on the observation that fourth differences can be constructed by subtraction of two second-order-accurate representations of the Laplacian with the same functional form of the leading term in the truncation error, but with different truncation error constants.

The first Laplacian is computed from

$$L(\phi)_0^{(\mathrm{I})} \approx \sum_{0i \in \mathcal{E}_0} (\phi_i - \phi_0) \tag{7}$$

The second Laplacian is approximated as

$$L(\phi)_0^{(\mathrm{II})} \approx \sum_{0i \in \mathcal{E}_0} \frac{1}{2}[(\nabla\phi)_0 + (\nabla\phi)_i] \cdot \Delta \boldsymbol{r}_{0i} \tag{8}$$

where $\Delta \boldsymbol{r}_{0i} = \boldsymbol{r}_i - \boldsymbol{r}_0$, with $\boldsymbol{r}$ being the position vector.

On uniform triangular and quadrilateral grids, both are second-order-accurate discretizations of the Laplacian operator. On a uniform quadrilateral grid with grid-spacing unity, the two leading terms in the truncation errors are given by

$$\mathrm{TE}^{(\mathrm{I})} = \frac{1}{12}\left(\frac{\partial^4\phi}{\partial x^4} + \frac{\partial^4\phi}{\partial y^4}\right) + \frac{1}{360}\left(\frac{\partial^6\phi}{\partial x^6} + \frac{\partial^6\phi}{\partial y^6}\right)$$

for the first Laplacian and

$$\mathrm{TE}^{(\mathrm{II})} = \frac{1}{3}\left(\frac{\partial^4\phi}{\partial x^4} + \frac{\partial^4\phi}{\partial y^4}\right) + \frac{2}{45}\left(\frac{\partial^6\phi}{\partial x^6} + \frac{\partial^6\phi}{\partial y^6}\right)$$

for the second Laplacian. Hence, by subtraction of the first discretization of the Laplacian from the second, a second-order-accurate discretization of (a quarter) of the fourth derivatives is obtained. The stencil that arises is shown in Fig. 2b. The Taylor series analysis of the second stencil (with the weights multiplied by a factor of four) gives

$$\frac{\partial^4\phi}{\partial x^4} + \frac{\partial^4\phi}{\partial y^4} + \frac{1}{6}\left(\frac{\partial^6\phi}{\partial x^6} + \frac{\partial^6\phi}{\partial y^6}\right) + \mathrm{HOT}$$

The second stencil corresponds to that commonly used on structured grids. It is noted that, because of the mixed derivatives, the stencil obtained with the first method is larger.

With the second method for computing the fourth differences, the final form of the artificial dissipation term for node 0 is

$$\boldsymbol{D}_0 = \sum_{0i \in \mathcal{E}_0} \Lambda_{0i}\varepsilon_{0i}^{(2)}(\boldsymbol{W}_i - \boldsymbol{W}_0) - \sum_{0i \in \mathcal{E}_0} 4\Lambda_{0i}\varepsilon_{0i}^{(4)}\left\{(\boldsymbol{W}_i - \boldsymbol{W}_0) - \frac{1}{2}[(\nabla\boldsymbol{W})_0 + (\nabla\boldsymbol{W})_i] \cdot \Delta\boldsymbol{r}_{0i}\right\} \tag{9}$$

It is interesting to note that this method of constructing fourth differences can also be cast in the framework of upwind schemes based on Riemann solvers that use left and right states to compute inviscid fluxes. At the interface $0i$, this can be achieved by

$$\boldsymbol{W}_{0i,L} - \boldsymbol{W}_{0i,R} = \left[\boldsymbol{W}_0 + \tfrac{1}{2}(\nabla\boldsymbol{W})_0 \cdot \Delta\boldsymbol{r}_{0i}\right] - \left[\boldsymbol{W}_i - \tfrac{1}{2}(\nabla\boldsymbol{W})_i \cdot \Delta\boldsymbol{r}_{0i}\right]$$

which allows the final form of the artificial dissipation term for node 0 to be re-expressed as

$$\boldsymbol{D}_0 = \sum_{0i \in \mathcal{E}_0} \Lambda_{0i}\varepsilon_{0i}^{(2)}(\boldsymbol{W}_i - \boldsymbol{W}_0) - \sum_{0i \in \mathcal{E}_0} 4\Lambda_{0i}\varepsilon_{0i}^{(4)}(\boldsymbol{W}_{0i,L} - \boldsymbol{W}_{0i,R})$$

It will be shown that the differences in the stencil weights have a considerable influence on the behavior of the JST scheme on various grid types.

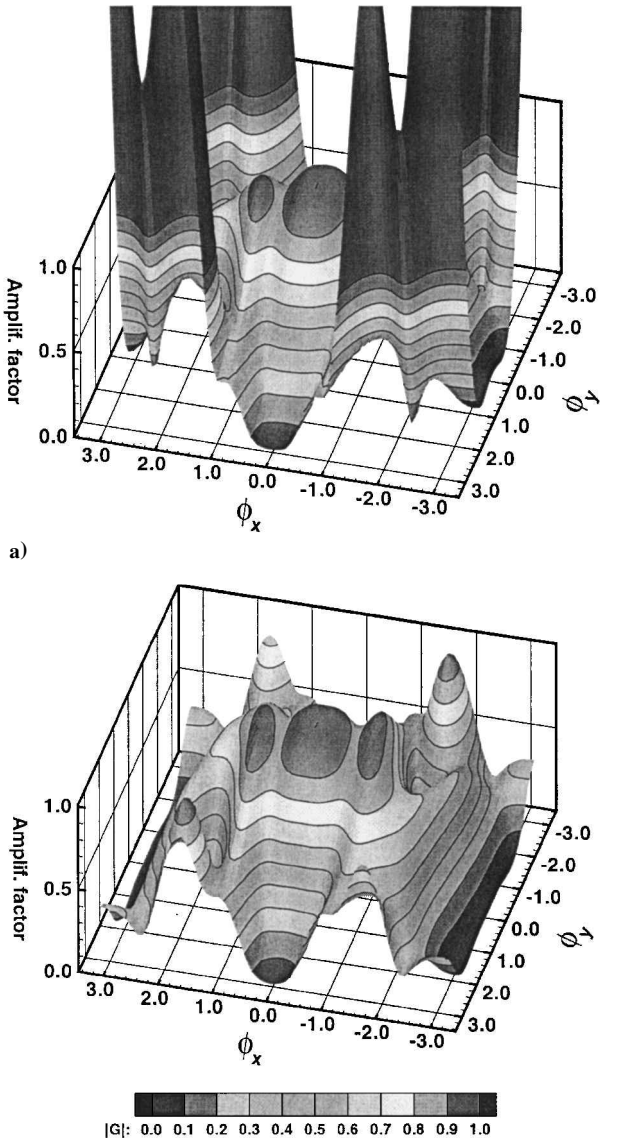


Fig. 3 Amplification factors for a) first method and b) second method of constructing fourth differences with $k^{(2)} = 0$ and $k^{(4)} = \frac{1}{32}$ on a quadrilateral grid for a five-stage Runge–Kutta scheme with two evaluations of dissipation.

Fourier Analysis

The goal of using fourth-difference operators is the damping of high-frequency solution modes. To assess the damping properties of the two fourth-difference operators, a Fourier analysis was carried out for the convection equation with strong flow-to-grid alignment as is typically encountered in the prismatic regions of mixed grids.

The resulting amplification factors for the two methods are shown in Fig. 3 for $k^{(2)} = 0$ and $k^{(4)} = \frac{1}{32}$ and a five-stage Runge–Kutta scheme with evaluations of dissipation in the first two stages. It is seen that the amplification factor for the first method of constructing fourth differences is larger than unity for high-frequency modes and hence unstable. In contrast, the amplification factor for the second method is below unity for high-frequency modes. This observation will be shown to be consistent with the results of a numerical study described below.

Although $k^{(4)} = \frac{1}{32}$ is admittedly somewhat higher than values usually used, the comparison serves to illustrate that discretizations developed on one grid type (in this case, triangular grids) cannot a priori be expected to work well on another grid type (in this case, quadrilateral grids). Furthermore, somewhat higher values of $k^{(4)}$ are sometimes used in the industrial environment to ensure fast convergence because the correct prediction of changes is more important than absolute accuracy.

Upwind Scheme

If the spectral radii in Eq. (9) were replaced with the Jacobian matrix of the inviscid fluxes, a matrix-dissipation scheme[14] would be obtained. The matrix-dissipation scheme may be regarded as the link between the JST scheme with scalar dissipation and upwind schemes such as Roe's approximate Riemann solver.[11]

To compute the convective flux with Roe's approximate Riemann solver, left and right states have to be defined at control-volume faces. For a second-order accurate scheme,

$$\phi_{0i,L} = \phi_0 + \tfrac{1}{2}\lambda_0(\nabla\phi)_0 \cdot \Delta\boldsymbol{r}_{0i}, \qquad \phi_{0i,R} = \phi_i - \tfrac{1}{2}\lambda_i(\nabla\phi)_i \cdot \Delta\boldsymbol{r}_{0i} \tag{10}$$

where λ_0 denotes the limiter function at node 0. In this paper the limiter function of Venkatakrishnan[15] is used. The measure of the local grid spacing required in Venkatakrishnan's limiter function is taken as the cube root of the control volume for which the limiter is computed, and the value of the constant is equal to 5. The second-order-accurate upwind scheme requires the reconstruction of gradients, which is described next.

Green–Gauss Gradient Reconstruction

Barth and Jespersen[16] suggested the use of a discrete version of the Green–Gauss theorem to reconstruct gradients. The resulting formula may be written as[16]

$$(\nabla\phi)_0 \approx \frac{1}{\Omega_0}\oint_{\partial\Omega_0}\frac{1}{2}(\phi_0 + \phi_i)\boldsymbol{n}_{0i}S_{0i} \tag{11}$$

which gives exact gradients for linear functions on tetrahedra only and is therefore not well suited to mixed grids. This is a serious drawback because prisms are usually used near solid walls and because the accurate evaluation of gradients at walls is particularly important for heat transfer simulations.

It is possible to derive a form of the Green–Gauss reconstruction that gives exact gradients on all cell types, see, e.g., Refs. 17 and 18. This form involves all the vertices of the cells meeting at vertex 0 and not just the edge neighbors. To include the contributions of the vertices not linked by an edge to vertex 0, additional connectivity arrays or cell-based data structures are required. Both lead to substantial additional memory requirements, and the latter is not grid transparent. For these reasons, it was decided to use approximation (11) regardless of cell types, as done by other authors, e.g., Khawaja et al.[3]

To quantify the errors incurred by the Green–Gauss reconstruction, gradients of specified linear variations of the primitive variables on a given mixed grid were reconstructed. Figure 4 illustrates the percentage of errors. Note that the errors are particularly at the interface between prisms and tetrahedra. In the best case, the Green–Gauss method was observed to lead to kinks in the profiles of the thermodynamic variables in the immediate vicinity of solid walls. In the worst case, the inaccurate gradients lead to divergence of Roe's approximate Riemann solver. The use of limiter functions typically delayed, but could not prevent, divergence.

Least-Squares Gradient Reconstruction

The linear least-squares reconstruction has the advantage that it gives exact gradients for linear functions on arbitrary cells and is therefore well suited to mixed grids. Anderson and Bonhaus[19] presented a linear least-squares reconstruction method in two dimensions without derivation. In this paper the least-squares reconstruction method is extended to three dimensions.

The least-squares reconstruction method is based on the use of a first-order Taylor series approximation for each edge that is incident to node 0. With this procedure, the gradient at node 0 follows from the weighted sum

$$(\nabla\phi)_0 = \sum_{0i\,\in\,\mathcal{E}_0}\begin{Bmatrix}\alpha_{0i,1} - \dfrac{r_{12}}{r_{11}}\alpha_{0i,2} + \Psi\alpha_{0i,3}\\ \alpha_{0i,2} - \dfrac{r_{23}}{r_{22}}\alpha_{0i,3}\\ \alpha_{0i,3}\end{Bmatrix}\Delta\phi_{0i} \tag{12}$$

where $\Delta(\cdot)_{0i} = (\cdot)_i - (\cdot)_0$. The terms in Eq. (12) are given by

$$\alpha_{0i,1} = \frac{\Delta x_{0i}}{r_{11}^2}, \qquad \alpha_{0i,2} = \frac{1}{r_{22}^2}\left(\Delta y_{0i} - \frac{r_{12}}{r_{11}}\Delta x_{0i}\right)$$

$$\alpha_{0i,3} = \frac{1}{r_{33}^2}\left(\Delta z_{0i} - \frac{r_{23}}{r_{22}}\Delta y_{0i} + \Psi\Delta x_{0i}\right)$$

where

$$\Psi = \frac{r_{12}r_{23} - r_{13}r_{22}}{r_{11}r_{22}} \tag{13}$$

$$r_{11} = \sqrt{\sum_{0i\,\in\,\mathcal{E}_0}\Delta^2 x_{0i}}, \qquad r_{12} = \frac{1}{r_{11}}\sum_{0i\,\in\,\mathcal{E}_0}\Delta x_{0i}\Delta y_{0i}$$

$$r_{13} = \frac{1}{r_{11}}\sum_{0i\,\in\,\mathcal{E}_0}\Delta x_{0i}\Delta z_{0i}, \qquad r_{22} = \sqrt{\sum_{0i\,\in\,\mathcal{E}_0}\Delta^2 y_{0i} - r_{12}^2}$$

$$r_{23} = \frac{1}{r_{22}}\left(\sum_{0i\,\in\,\mathcal{E}_0}\Delta y_{0i}\Delta z_{0i} - r_{12}r_{13}\right)$$

$$r_{33} = \sqrt{\sum_{0i\,\in\,\mathcal{E}_0}\Delta^2 z_{0i} - \left(r_{13}^2 + r_{23}^2\right)}$$

The linear least-squares gradient-reconstruction method was found to improve the robustness of the solution method significantly. The improvement comes at the cost of having to store the six floating-point values at each vertex.

Virtual Edges

If prismatic or hexahedral cells are used on a flat solid boundary, the gradient normal to the wall will be constructed from a single edge. To increase the support of the stencil used to reconstruct gradients, so-called virtual edges are inserted along the diagonals of those quadrilateral faces that touch solid walls, as shown in Fig. 5a. These additional edges are termed virtual edges because they are used only to reconstruct gradients, in contrast to the actual edges in the grid. Virtual edges are also inserted into prismatic, pyramidal, and hexahedral cells at cell-type interfaces to ensure that the stencil is more isotropic, as illustrated in Fig. 5b for two dimensions.

Discretization of Viscous Fluxes

For the discretization of the viscous fluxes, the first derivatives are required at the edge midpoints. With the edge-based data structure, it is tempting to compute the first derivatives at edge midpoints by a simple average:

$$\overline{(\nabla\phi)}_{0i} = \tfrac{1}{2}[(\nabla\phi)_0 + (\nabla\phi)_i] \tag{14}$$

The stencils that are obtained on triangular and quadrilateral grids with this method for the discretization of the Laplacian (as a model for the viscous terms) are shown in Fig. 6. It can be seen that decoupling occurs for quadrilateral grids. This is particularly significant because the grids near solid boundaries will be either hexahedral or prismatic, both of which contain quadrilateral faces. Decoupling is undesirable because it allows spurious solution modes and leads to larger truncation errors.

The decoupling on quadrilateral and hexahedral grids can be prevented by use of the directional derivative along the edge:

$$\left(\frac{\partial\phi}{\partial l}\right)_{0i} = \frac{\phi_i - \phi_0}{l_{0i}} \tag{15}$$

where $l_{0i} = \|\Delta\boldsymbol{r}_{0i}\|$ denotes the length of edge $0i$. With the definition of the unit vector $\boldsymbol{t}_{0i}$ along edge $0i$,

$$\boldsymbol{t}_{0i} = \Delta\boldsymbol{r}_{0i}/l_{0i} \tag{16}$$

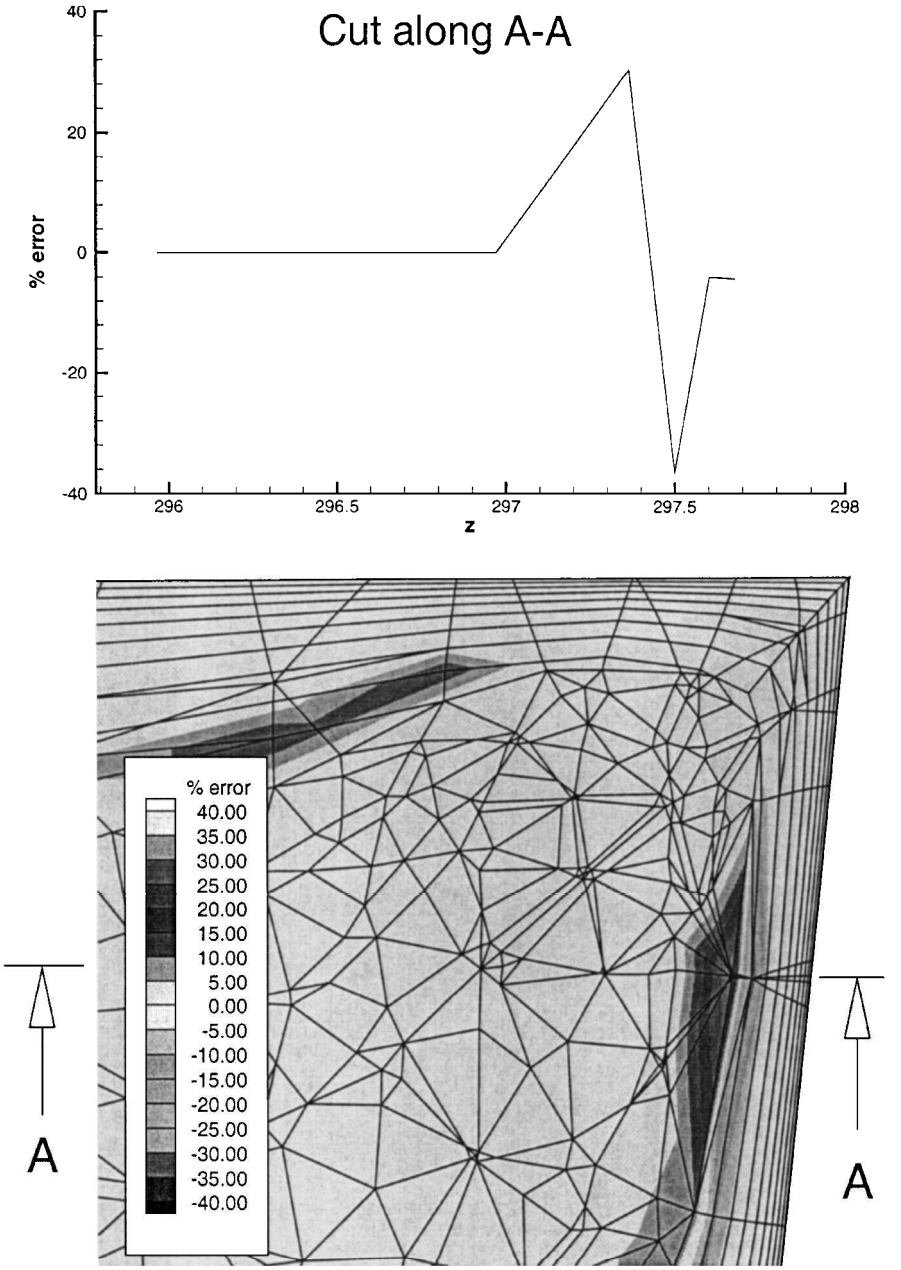


Fig. 4 Percentage of error of the Green–Gauss reconstruction of the magnitude of the first derivative of a linear function on a mixed grid.

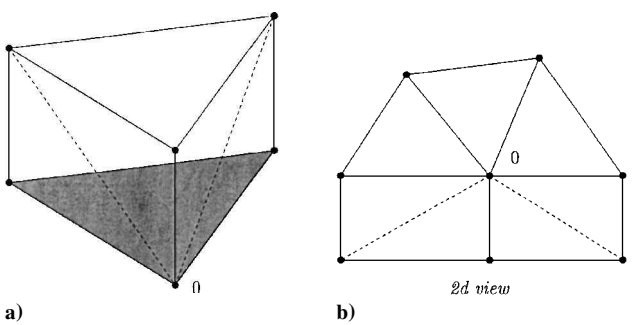


Fig. 5 Virtual edges (dashed lines) at node 0 for a) a prism cell on boundary and (boundary face shown shaded) and b) quadrilateral cells at interfaces [two-dimensional (2d) view shown for simplicity].

the modified average that prevents decoupling may be written as

$$(\nabla\phi)_{0i} = \overline{(\nabla\phi)}_{0i} - \left[\overline{(\nabla\phi)}_{0i} \cdot \boldsymbol{t}_{0i} - \left(\frac{\partial\phi}{\partial l}\right)_{0i}\right]\boldsymbol{t}_{0i} \qquad (17)$$

With the modified average, strongly coupled stencils are obtained, as shown in Fig. 7. The modified approach is still compatible with the edge-based data structure and requires no additional storage. This approach was used by a number of other authors; see, e.g., Weiss et al.[20]

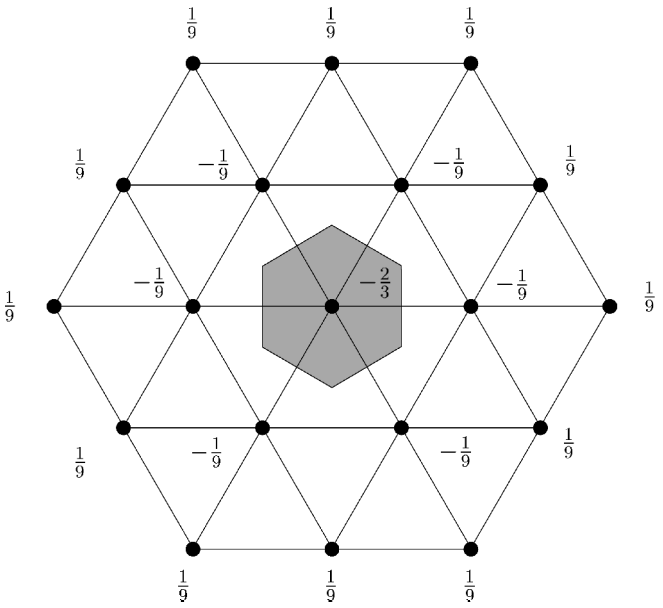
Results

Inviscid Flow

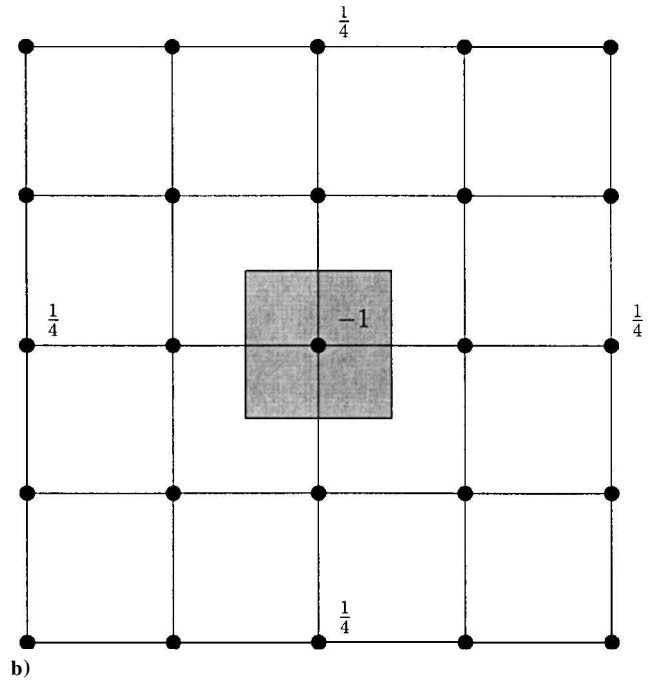
To illustrate the influence of the two stencils for the fourth-difference operator in the JST scheme, the transonic flow over a circular-arc bump in a straight-walled channel is computed. A two-dimensional code for unstructured mixed grids, described in Ref. 5, was used. A triangular grid of 3276 vertices and 6350 cells and a quadrilateral grid of 3332 vertices and 3201 cells are used for the comparison. The grids are shown in Fig. 8.

Table 1 Comparison of storage requirements for floating-point quantities specific to central and upwind schemes for Euler and Navier–Stokes (NS) equations

Scheme	Euler with limiters	NS with limiters	NS without limiters
Central 1	11	23	21
Central 2	27	27	22
Upwind	36	36	21

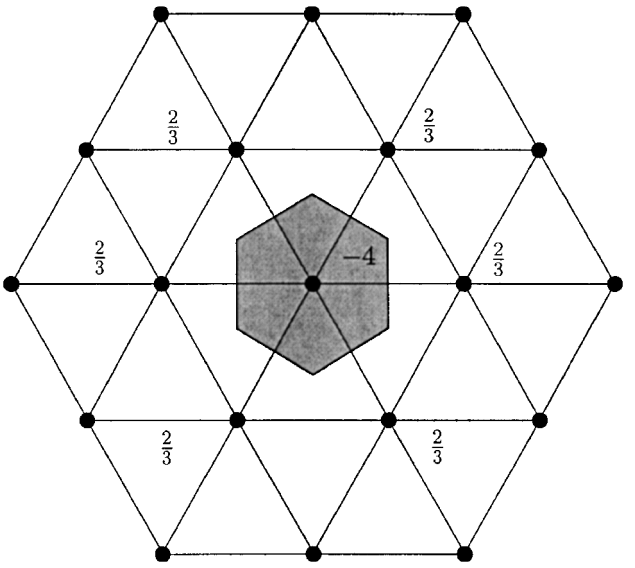


a)

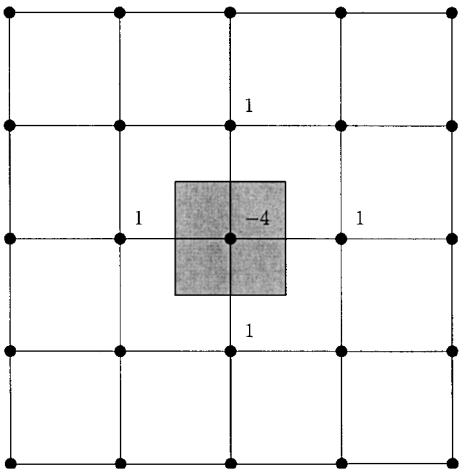


b)

Fig. 6 Stencils obtained when node gradients averaged along the edge for a) equilateral triangular grid and b) uniform quadrilateral grid.

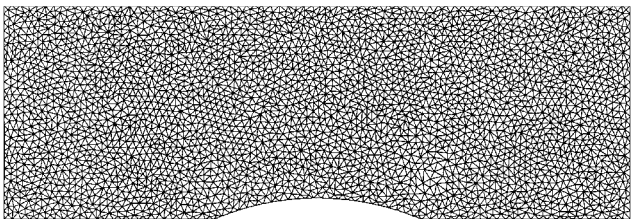


a)

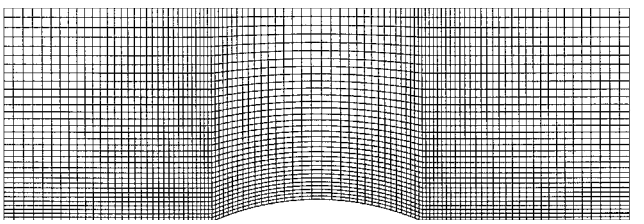


b)

Fig. 7 Stencils obtained with modified averaged gradient for a) equilateral triangular grid and b) uniform quadrilateral grid.

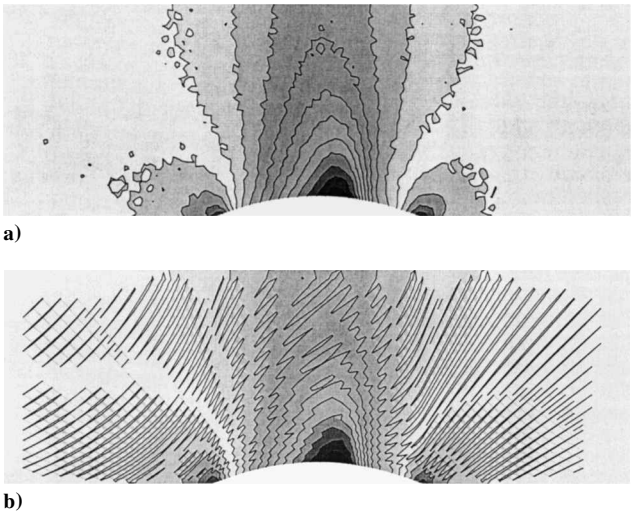


a)



b)

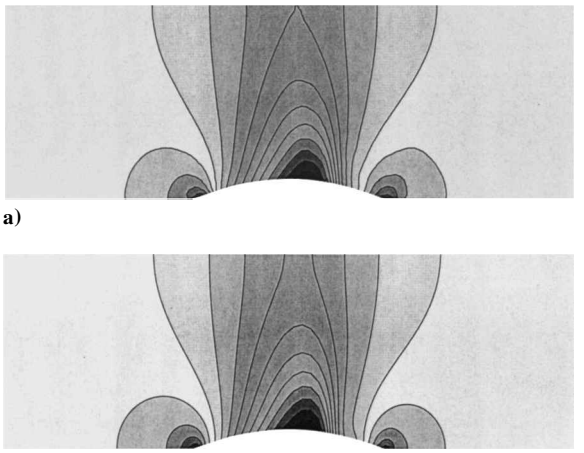
Fig. 8 Grids used for comparison of fourth-difference operators for transonic inviscid flow: a) triangular and b) quadrilateral.



a)

b)

Fig. 9 Pressure contours for first method of constructing fourth differences with $k^{(4)} = \frac{1}{48}$ on a) triangular grid and b) quadrilateral grid.



a)

b)

Fig. 10 Pressure contours for second method of constructing fourth differences with $k^{(4)} = \frac{1}{32}$ on a) triangular grid and b) quadrilateral grid.

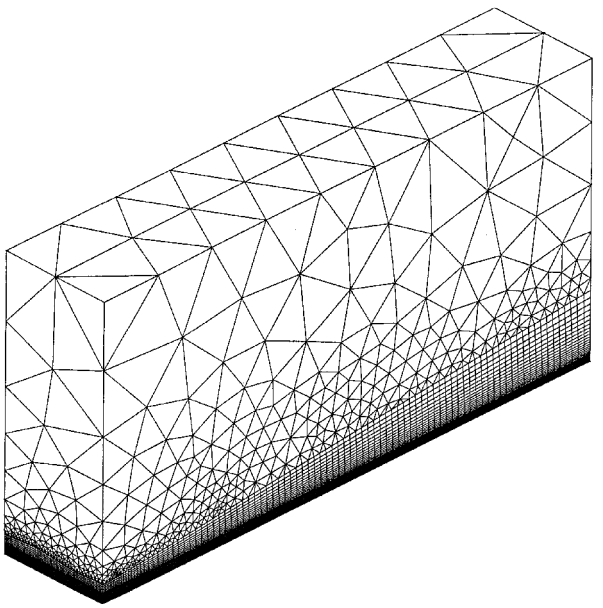
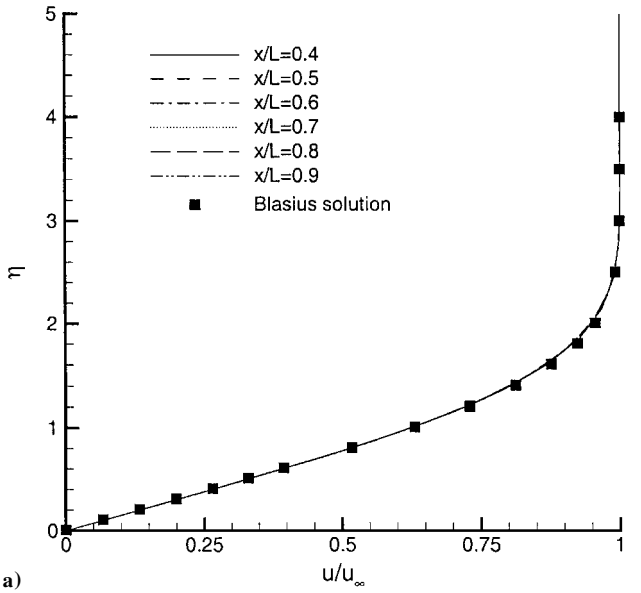


Fig. 11 Mixed grid used for computing laminar boundary layer on a flat plate.

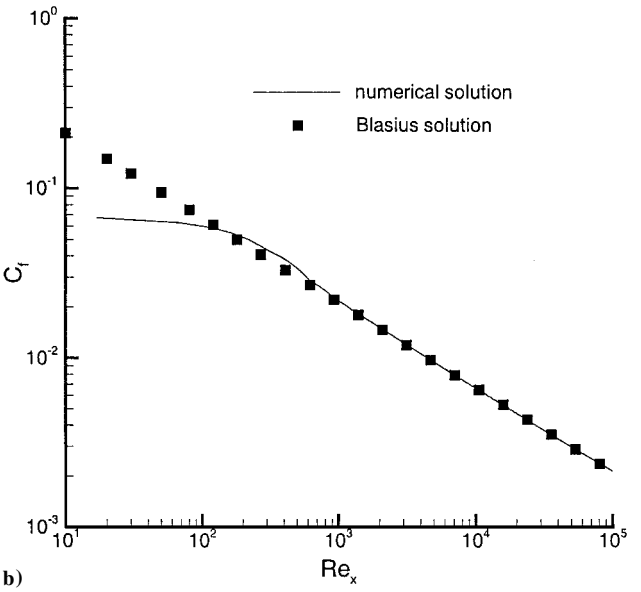
To compare the two forms of constructing fourth-difference operators, computations were carried out for $\frac{1}{32} \leq k^{(4)} \leq \frac{1}{2048}$. Sample solutions are shown in Figs. 9 and 10, which demonstrate that the first method shows signs of instability. For $k^{(4)} = \frac{1}{32}$, the first method diverged on triangular grids. These results are consistent with the Fourier analysis. The convergence histories for the two methods on the triangular and quadrilateral grids can be found in Ref. 21.

Corresponding calculations in three dimensions showed similar behavior. The range of values of $k^{(4)}$, however, was much reduced. This unsatisfactory behavior motivated the change from the JST scheme to Roe's approximate Riemann solver.

The main penalty of switching to Roe's scheme is increased CPU time. The storage required is increased only if a limiter function is required, as demonstrated in Table 1. This table compares the storage requirements of the central scheme with the first and the second method of constructing fourth differences with the upwind scheme for inviscid and laminar flows with and without limiters. It can be seen that for viscous flows for which no limiter function is required, the upwind scheme and the central schemes require approximately the same amount of storage. This is because gradients reconstructed for the higher-order inviscid fluxes may be reused for the computation of the viscous fluxes, so that



a)



b)

Fig. 12 Comparison of a) normalized u velocity at various stations along plate and b) skin-friction coefficient with Blasius solution.

they do not incur a storage penalty. Because the present flow solution method is aimed at internal flows that are mostly subsonic, the switch to the upwind method does not lead to a storage penalty.

Laminar Flow

Flat Plate

The laminar flow over a flat plate is computed with the upwind scheme on the mixed grid shown in Fig. 11. The grid consists of 81,805 nodes, 22,899 tetrahedra, and 144,150 prisms.

The computed profiles of normalized u velocity are shown in Fig. 12a for various stations and compared with the exact solution. Excellent agreement with the exact solution is obtained.

The comparison of computed and exact variations of the skin-friction coefficient shows good agreement as well, as may be seen from Fig. 12b. The discrepancies near the inlet are due to the specification of a uniform velocity profile.

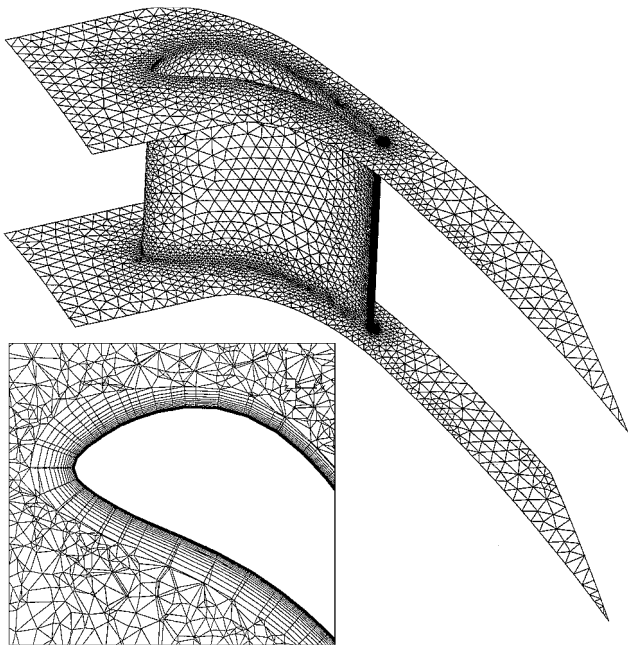


Fig. 13 Mixed grid used for computing laminar flow about turbine guide vane. The inset shows layers of prismatic cells around blade near leading edge.

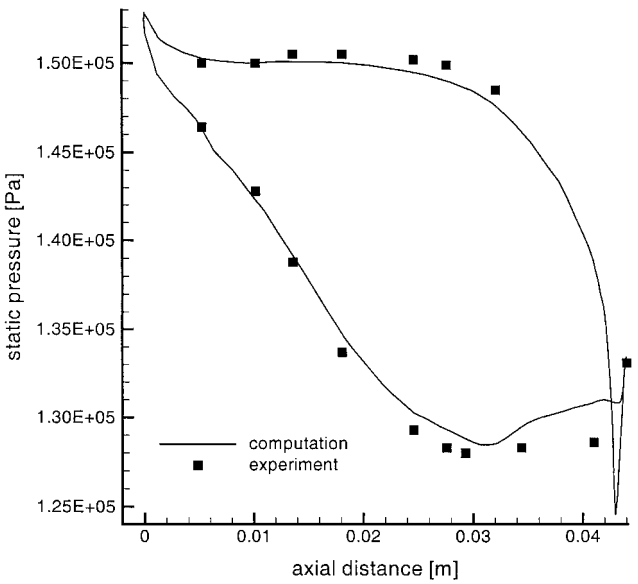


Fig. 14 Comparison of computed and experimental variations of static pressure at half blade height.

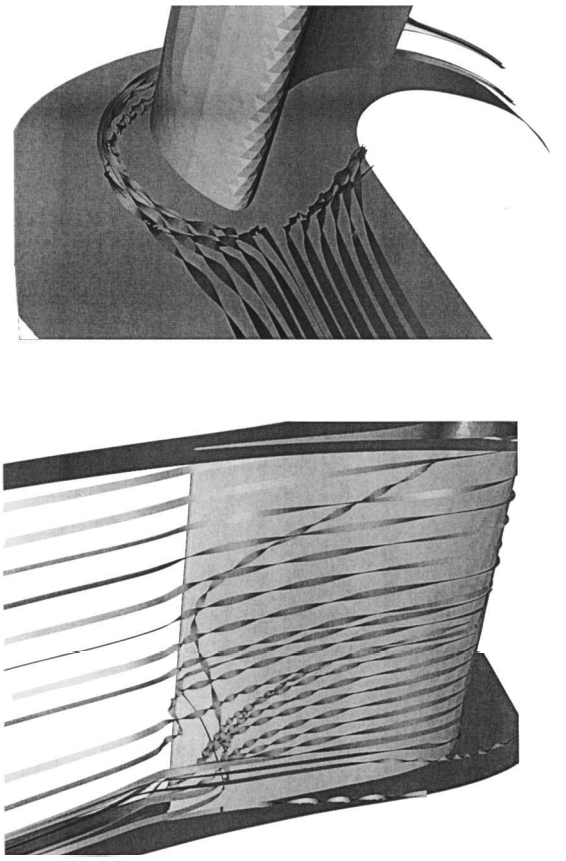


Fig. 15 Visualization of horseshoe vortex and separation bubble near trailing edge on suction side.

Turbine Guide Vane

The next test case considers the laminar flow about a turbine guide vane investigated experimentally by Zeschky.[22] The grid is shown in Fig. 13 and consists of 207,583 vertices, 453,030 tetrahedra, 231,587 prisms and 5050 pyramids.

The comparison of the computed variation of static pressure with the experimental values is shown in Fig. 14. The slight discrepancy on the pressure side may be due to a lack of grid resolution. The discrepancy near the trailing edge of the suction side may be due to failing to resolve the wake. The horseshoe vortex and the separation bubble (as observed in the experiment[20]) are shown in Fig. 15.

Conclusions

The discretization of the Navier–Stokes equations on mixed unstructured grids was discussed. Two methods of constructing fourth-difference dissipation operators for centered discretizations of the inviscid fluxes were compared theoretically and numerically in two dimensions. One form was shown to exhibit better properties. Problems with the construction of the fourth-difference operator remained in three dimensions. For this reason, the centered discretization was replaced with an upwind scheme, for which a previously derived gradient-reconstruction method was extended to three dimensions.

Results for a laminar boundary layer on a flat plate showed excellent agreement with the theoretical solution. Good agreement with experimental results was obtained for the laminar flow about a turbine guide vane.

Acknowledgments

The authors would like to acknowledge Dimitri Mavriplis for his observation that the second construction of fourth-order differences can be written in the form of an upwind scheme. The grids for the flat plate and the turbine guide vane were generated with CENTAUR™; see http://www.centaursoft.com.

References

[1]Frink, N. T., Parikh, P., and Pirzadeh, S., "A Fast Upwind Solver for the Euler Equations on Three-Dimensional Unstructured Meshes," AIAA Paper 91-0102, Jan. 1991.

[2]Mavriplis, D. J., and Venkatakrishnan, V., "A Unified Multigrid Solver for the Navier–Stokes Equations on Mixed Element Meshes," Inst. for Computer Application in Science and Engineering, ICASE Rept. 95-53, Hampton, VA, July 1995.

[3]Khawaja, A., Kallinderis, Y., and Parthasarathy, V., "Implementation of Adaptive Hybrid Grids for 3-D Turbulent Flows," AIAA Paper 96-0026, Jan. 1996.

[4]Coirier, W. J., and Jorgenson, P. C. E., "A Mixed-Volume Approach for the Euler and Navier–Stokes Equations," AIAA Paper 96-0762, Jan. 1996.

[5]Haselbacher, A., McGuirk, J. J., and Page, G. J., "Finite-Volume Discretization Aspects for Viscous Flows on Mixed Unstructured Grids," *AIAA Journal*, Vol. 37, No. 2, 1999, pp. 177–184.

[6]Blazek, J., Irmisch, S., and Haselbacher, A., "Unstructured Mixed-Grid Navier–Stokes Solver for Turbomachinery Applications," AIAA Paper 99-0664, Jan. 1999.

[7]Shaw, J. A., "Hybrid Grids," Aircraft Research Association, TM 426, Bedford, England, U.K., March 1998.

[8]Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for the Euler and Navier–Stokes Equations," *Special Course on Unstructured-Grid Methods for Advection-Dominated Flows*, Rept. 787, AGARD, May 1992, pp. 6-1–6-61.

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

[10]Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA Paper 86-0103, Jan. 1986.

[11]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[12]Holmes, G. D., and Connell, S. D., "Solution of the 2-D Navier–Stokes Equations on Unstructured Grids," AIAA Paper 89-1932, June 1989.

[13]Frink, N. T., "Recent Progress Towards a Three-Dimensional Unstructured Navier–Stokes Flow Solver," AIAA Paper 94-0061, Jan. 1994.

[14]Swanson, R. C., and Turkel, E., "On Central-Difference and Upwind Schemes," *Journal of Computational Physics*, Vol. 101, No. 2, 1992, pp. 292–306.

[15]Venkatakrishnan, V., "Convergence to Steady-State of the Euler Equations on Unstructured Grids with Limiters," *Journal of Computational Physics*, Vol. 118, No. 1, 1995, pp. 120–130.

[16]Barth, T. J., and Jespersen, D., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0366, Jan. 1989.

[17]Müller, J.-D., and Giles, M. B., "Edge-Based Multigrid Schemes for Hybrid Grids," *Proceedings of the 6th ICFD Conference on Numerical Methods for Fluid Dynamics*, Inst. for Computational Fluid Dynamics, Oxford Univ., Oxford, 1998, pp. 425–432.

[18]Haselbacher, A., "A Grid-Transparent Numerical Method for the Computation of Compressible Viscous Flows on Mixed Unstructured Grids," Ph.D. Dissertation, Loughborough Univ., Loughborough, Leicestershire, England, U.K., Jan. 2000.

[19]Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

[20]Weiss, J. M., Maruszewski, J. P., and Smith, W. A., "Implicit Solution of Preconditioned Navier–Stokes Equations Using Algebraic Multigrid," *AIAA Journal*, Vol. 37, No. 1, 1999, pp. 29–36.

[21]Haselbacher, A., and Blazek, J., "On the Accurate and Efficient Discretisation of the Navier–Stokes Equations on Mixed Grids," AIAA Paper 99-3363, June/July 1999.

[22]Zeschky, J., "Experimentelle Untersuchung der dreidimensionalen instationären Rotorströmung in einer axialen Kaltluftturbine," Ph.D. Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen, Aachen, Germany, July 1991.

K. Kailasanath
Associate Editor