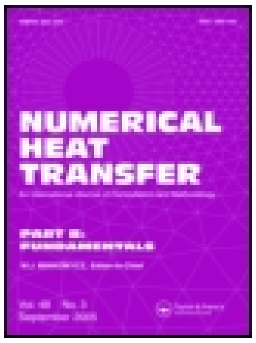


Numerical Heat Transfer

ISSN: 1040-7790 (Print) 1521-0626 (Online) Journal homepage: https://www.tandfonline.com/loi/unhb20

# A PRESSURE-BASED METHOD FOR UNSTRUCTURED MESHES

S. R. Mathur & J. Y. Murthy

**To cite this article:** S. R. Mathur & J. Y. Murthy (1997) A PRESSURE-BASED METHOD FOR UNSTRUCTURED MESHES, Numerical Heat Transfer, 31:2, 195-215, DOI: 10.1080/10407799708915105

**To link to this article:** https://doi.org/10.1080/10407799708915105

# A PRESSURE-BASED METHOD FOR UNSTRUCTURED MESHES

*S. R. Mathur and J. Y. Murthy*
*Fluent, Inc., 10 Cavendish Court, Lebanon, New Hampshire 03766, USA*

*This article presents a finite-volume scheme for multidimensional incompressible flows. Unstructured, solution-adaptive meshes composed of arbitrary convex polyhedra are used. A cell-centered equal-order formulation is developed. Gradients required for the evaluation of diffusion fluxes and for second-order-accurate convective operators are found by linear reconstruction. An additive-correction multigrid scheme is used to solve the resulting discrete equations. Pressure and velocity are stored at cell centers; momentum interpolation is used to prevent pressure checkerboarding. The SIMPLE algorithm is used for pressure–velocity coupling. Schemes for hanging-node and conformal adaption are implemented. The scheme is applied to benchmark problems using a variety of quadrilateral/hexahedral, triangular/tetrahedral, and hybrid meshes, and is shown to perform satisfactorily.*

## INTRODUCTION

Unstructured grid methods have been the focus of a considerable amount of computational fluid dynamics research over the last 10–15 years. The two main advantages of these methods, viz., the ease of grid generation for complex, realistic geometries, and the ability to dynamically adapt the grid to local features of interest, have the potential of establishing numerical simulation as an integral component of design and optimization cycles.

The aerospace industry was among the first to benefit from advances in this field. Jameson et al. [1] were able to compute flow fields around complex configurations, such as the Boeing 747 aircraft, using Euler equations on tetrahedral meshes. The high-subsonic to fully supersonic flow regimes of interest meant that the density-based, explicit time-marching algorithms, used in conjunction with multigrid acceleration, were competitive in performance with traditional structured grid-based approaches. These methods have been extended to Reynolds-averaged Navier-Stokes equations in recent years with some success [2]. Attempts have also been made to extend the time-marching framework to handle the incompressible and low-subsonic regimes. These have ranged from the simple pseudo-compressibility [3, 4] concept to more sophisticated preconditioning algorithms [5, 6]. Their use in industrial-scale applications, however, has been limited, since the

Received 1 July 1996; accepted 5 August 1996.
The authors acknowledge the contributions of B. Makarov, J. Maruszewski, and G. Spragle of Fluent, Inc., to various aspects of this work. The use of Fluent, Inc.'s, software, FLUENT/UNS and PreBFC, is also gratefully acknowledged.
Address correspondence to Dr. S. R. Mathur, Fluent, Inc., 10 Cavendish Court, Lebanon, NH 03766, USA. E-mail: sm@fluent.com

## NOMENCLATURE

| | | | |
|---|---|---|---|
| $\mathbf{A}$ | area vector | $T$ | temperature |
| $A_i$ | components of $\mathbf{A}$ | $u_i$ | velocity component in Cartesian direction $i$ |
| $C_p$ | specific heat at constant pressure | | |
| $D_f$ | diffusion flux on face $f$ | $x_i$ | coordinate direction |
| $F_i$ | body force in direction $i$ | $\Delta\mathcal{V}$ | volume of control volume |
| $g$ | acceleration due to gravity | $\Gamma$ | diffusion coefficient |
| $J_f$ | mass flow rate on face $f$ | $\mu$ | molecular viscosity |
| Nu | Nusselt number | $\nu$ | kinematic viscosity |
| $p$ | static pressure | $\rho$ | density |
| $S_h$ | energy source per unit volume | $\tau_{ij}$ | stress tensor |
| $S_\phi$ | source of $\phi$ per unit volume | $\phi$ | transported scalar |

small grid sizes that are necessary to resolve viscous features often impose severe time-step restrictions on explicit schemes. Coupled, implicit algorithms, on the other hand, are much more expensive in storage and CPU requirements [7] compared to their structured grid counterparts, since practices such as approximate factorization cannot be employed.

Pressure-based methods have been the methods of choice for incompressible flows over the last two decades. The combined use of a finite-volume discretization with a segregated solution strategy and pressure–velocity coupling via a pressure-correction equation make algorithms such as SIMPLE [8] and its extensions among the most efficient approaches for incompressible flows. However, extension of this well-known methodology to unstructured grids has been hampered by two main difficulties. Staggered storage of velocity and pressure components, necessary to avoid checkerboarding, is not straightforward. Also, gradient determination is complicated by the absence of line structure. This leads to difficulties in discretization of diffusion fluxes and the development of higher-order schemes.

A variety of finite-element methods have been developed for incompressible flows. These eliminate the dependence on line structure through the use of local shape functions. Pressure checkerboarding is eliminated through the use of unequal-order interpolation [9, 10]; equal-order interpolation has also been used [11]. Higher-order discretization is achieved through the use of higher-order elements, with convective terms stabilized through the use of schemes such as streamline upwind Petrov-Galerkin (SUPG) [12].

Attempts have been made to combine the geometric flexibility of finite-element methods with the conservative property of finite-volume methods through the development of control-volume finite-element methods (CVFEM) [13, 14]. Here, variables are stored at element vertices, with either unequal-order [13] or equal-order [14, 15] interpolation of pressure and velocity. Higher-order discretization has been achieved through the use of streamwise exponential shape functions [13] or through element-based streamline upwinding [14]. Recently, unstructured cell-based finite-volume schemes are beginning to appear [16–18], which are more closely related to traditional methods for structured body-fitted meshes [19, 20]. Here conservation is enforced on the basic cell itself, and not on the cell-dual, as in [13]. Co-located pressure–velocity storage has been used in [16–18]. A staggered mesh-like scheme has been developed in [21]. Since gradient determination does not employ element-specific shape functions, these methods have the potential for use with arbitrary polyhedral meshes.

In this article we develop a pressure-based finite-volume scheme for unstructured meshes similar in philosophy to [16–18]. The method admits arbitrary convex polyhedra, including meshes with hanging nodes. Cell-based, colocated storage is preferred, and the use of element-specific shape functions is avoided. Discretization operators for the convective and diffusive fluxes are formulated so as to reduce to the well-known forms on body-fitted structured grids. Higher-order fluxes as well as secondary diffusion terms are computed using linear reconstruction and limiting similar to methods used for compressible flows. To minimize storage requirements, a segregated solution strategy is favored, with pressure and velocity coupled using the SIMPLE algorithm. An algebraic multigrid solver is used for the solution of linearized equations. The present method thus combines the attractive features of pressure-based schemes with advances made in other unstructured grid techniques.

The next two sections present the governing equations and details of the numerical method respectively. For the sake of clarity, only the steady, laminar, and incompressible flow of a constant property fluid is considered, although the method is easily extended to unsteady, compressible, and turbulent regimes.

## GOVERNING EQUATIONS

The equations for conservation of mass, momentum, and energy are

$$\frac{\partial}{\partial x_i}(\rho u_i) = 0 \tag{1}$$

$$\frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + F_i \tag{2}$$

$$\frac{\partial}{\partial x_i}(\rho u_i C_p T) = \frac{\partial}{\partial x_i}\left(k\frac{\partial T}{\partial x_i}\right) + (\tau_{ij})\frac{\partial u_i}{\partial x_j} + S_h \tag{3}$$

A Newtonian fluid is assumed. Thus

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial u_l}{\partial x_l}\delta_{ij} \tag{4}$$

## NUMERICAL METHOD

The domain is discretized into arbitrary unstructured convex polyhedra called cells. The boundaries surrounding the cells are called faces, and the vertices of the polyhedra are referred to as nodes. Each internal face has two cells on either side; these are referred to as the face neighbors. The neighbors of a cell are defined to be those cells with which it shares a common face. Line segments joining the nodes are termed "edges" (and are identical to faces in two dimensions). Meshes with "hanging nodes," such as the one shown in Figure 1*a*, are permitted.

All transport variables are stored at cell centers. This arrangement is preferred over node-based storage for several reasons. With cell-based storage, conservation can be ensured for arbitrary control volumes with nonconformal interfaces without special interpolation techniques. On triangular and tetrahedral meshes, cell-based storage also enjoys better resolution (since the ratio of number of cells to nodes is between 3 and 5) for roughly the same amount of work, which is typically proportional to the number of faces.

## Discretization of a Scalar Transport Equation

Consider the differential equation for the transport of a scalar quantity $\phi$. All the governing equations can be cast into the following form with the appropriate choice of $\phi$, $\Gamma$, and $S_\phi$:

$$\frac{\partial}{\partial x_i}(\rho u_i \phi) = \frac{\partial}{\partial x_i}\left(\Gamma \frac{\partial \phi}{\partial x_i}\right) + S_\phi \tag{5}$$

Integration and discretization about the control volume C0 shown in Figure 1*a* yields

$$\sum_f J_f \phi_f = \sum_f D_f + (S_\phi \Delta \mathscr{V})_0 \tag{6}$$

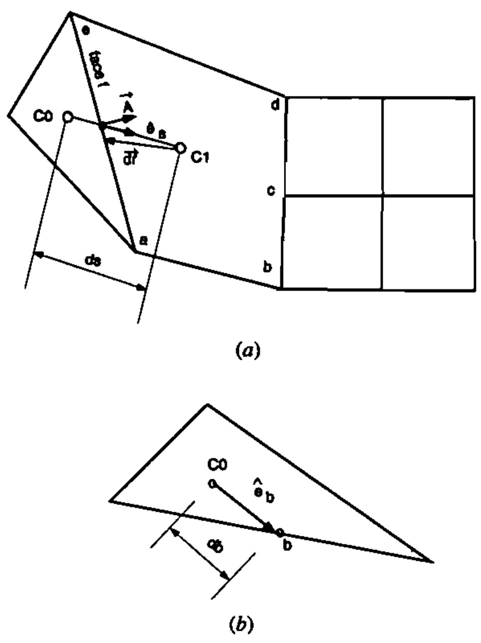


Figure 1. Control volume: (*a*) interior; (*b*) boundary.

where $J_f$ is the mass flow rate (defined to be positive if flow is leaving C0), $D_f$ is the transport due to diffusion through the face $f$, and the summations are over the faces of the control volume.

## Convection Term

The mass flow rate $J_f$ is assumed to be known from the solution of the momentum and continuity equations (see the following section). Therefore the task of evaluation of the convective flux reduces to determining the face value, $\phi_f$. A first-order approximation is to use the value at the upwind cell,

$$\phi_f = \phi_{\text{upwind}} \tag{7}$$

In structured grid methods, higher-order accuracy is achieved by employing a bigger stencil along the appropriate grid line. In the case of unstructured grids, however, this is not possible. Attempts have been made which mimic the bigger stencil by creating local line structures and interpolating values from neighboring cells [16]. These can be quite expensive and are also cell-shape-specific. However, if a shape-independent formulation for the cell gradients can be devised, a higher-order value at the face can be obtained as

$$\phi_f = \phi_{\text{upwind}} + \nabla\phi_{r_{\text{upwind}}} \cdot d\mathbf{r} \tag{8}$$

where $\nabla\phi_{r_{\text{upwind}}}$ is the *reconstruction gradient* at the upwind cell (see the section below) and $d\mathbf{r}$ is the vector directed from the centroid of the upwind cell to the centroid of the face. The first term in Eq. (8) is treated implicitly and the second term is included explicitly.

## Diffusion Term

The diffusion term at the face is

$$D_f = \Gamma_f \nabla\phi \cdot \mathbf{A} \tag{9}$$

We would like to express $D_f$ in terms of the values of $\phi$ at cells C0 and C1. To determine the form of this term, consider transformation from physical coordinates $(x, y)$ to computational coordinates $(\xi, \eta)$ shown in Figure 2. We can write

$$\nabla\phi \cdot \mathbf{A} = \phi_\xi(\xi_x A_x + \xi_y A_y) + \phi_\eta(\eta_x A_x + \eta_y A_y) \tag{10}$$

where $(A_x, A_y)$ are the Cartesian components of the area vector $\mathbf{A}$. As shown in the appendix, we may write Eq. (10) on the face $f$ as

$$\nabla\phi \cdot \mathbf{A} = \frac{\phi_1 - \phi_0}{ds}\frac{\mathbf{A}\cdot\mathbf{A}}{\mathbf{A}\cdot\hat{e}_s} + \frac{\phi_b - \phi_a}{A}\frac{\mathbf{A}\cdot\mathbf{A}}{\mathbf{A}\cdot\hat{e}_s}\hat{e}_t\cdot\hat{e}_s \tag{11}$$

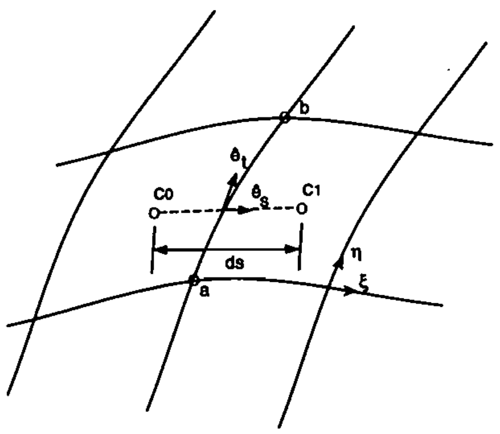


**Figure 2.** Structured grid control volume.

The discretization shown in Eq. (11) is identical to that used in structured-mesh schemes ([19], for example). The first term on the right-hand side of Eq. (11) represents the primary gradient; the discretization shown above is equivalent to a second-order central different representation. As in typical structured mesh formulations, this term is treated implicitly and leads to a stencil that includes all neighboring cells.

The second term is the secondary or cross-diffusion term and is zero when $\hat{e}_t \cdot \hat{e}_s$ is zero (e.g., on an orthogonal quadrilateral mesh or on a mesh of equilateral triangles). In general, however, this term can be significant, and its correct evaluation becomes important. In a structured grid formulation the node values $\phi_a$ and $\phi_b$ are typically found by averaging the values at cell centers and the entire secondary gradient component is treated explicitly. Similar treatment is easily possible in two dimensions on unstructured meshes. Jiang and Przekwas [16] have computed $\phi_a$ and $\phi_b$ using both inverse distance-weighted averaging and least-squares methods for the node values. Davidson [18] arrives at a similar formulation by constructing a parallelogram control volume about the face and also using node values for the secondary terms. However, extending this methodology to three-dimensional polyhedra of arbitrary shape is not straightforward. There are no unique face tangential directions and nodes that can be employed to write an equivalent form in three dimensions. (In the case of structured grids the underlying grid coordinates and nodes supply both the decomposition directions and the differencing stencils.)

To avoid the use of face tangents and nodes, the secondary diffusion term in Eq. (11) can be written as the difference between the total diffusion and the primary component. Thus

$$D_f = \Gamma_f \frac{(\phi_1 - \phi_0)}{ds} \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{e}_s} + \Gamma_f \left( \overline{\nabla \phi} \cdot \mathbf{A} - \overline{\nabla \phi} \cdot \hat{e}_s \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{e}_s} \right) \tag{12}$$

where $\overline{\nabla\phi}$ at the face is taken to be the average of the derivatives at the two adjacent cells. If the cell derivative is second-order-accurate, the diffusion term representation is second-order-accurate. As discussed in the next section, linear reconstruction can be used to determine $\nabla\phi$, the *cell derivative* of $\phi$, in a manner independent of cell shape. The formulation above is applicable without modification to three-dimensional configurations.

## Gradient Computation

In the absence of a line structure, the values of the variable and its derivatives cannot be written in terms of cell values using one-dimensional Taylor series expansions. CVFEM methods [13] overcome this by employing shape functions; another approach is to use interpolation methods such as least squares [22]. However, these techniques can be quite expensive and are cell-shape-specific. Linear reconstruction based on the divergence theorem provides a technique that is independent of cell shape [22]. The *reconstruction gradient* is estimated as

$$\nabla\phi_r = \frac{\alpha}{\Delta\mathscr{V}}\sum_f\left(\overline{\phi_f}\mathbf{A}\right) \tag{13}$$

where the summation is over all the faces of the cell. The face value of $\phi$ is obtained by averaging the values at the neighboring cells, so that for the face $f$ in Figure 1*a*,

$$\overline{\phi_f} = \frac{\phi_0 + \phi_1}{2} \tag{14}$$

$\alpha$ is a factor used to ensure that the reconstruction does not introduce local extrema. It is computed by first projecting the value of $\phi$ to all the nodes of the cell. The gradient at the cell is then limited such that the reconstructed value at each of the nodes is bounded by the maximum and minimum values of $\phi$ over all the cells that share the node. The limiter proposed by Venkatakrishnan [23] is used in the present work.

Using the reconstruction gradient, the value at any face of the cell can be reconstructed as

$$\phi_{f,c} = \phi_c + \nabla\phi_r \cdot d\mathbf{r} \tag{15}$$

For evaluating the convective flux at the face, the face value $\phi_f$ is taken to be the reconstructed value from the upwind cell. This results in an upwind-biased second-order representation of the convective term. The cell derivatives used for the secondary diffusion terms in Eq. (12) are computed by again applying the divergence theorem over the control volume and using the averaged reconstructed values at the faces,

$$\nabla\phi = \frac{1}{\Delta\mathscr{V}}\sum_f\left(\tilde{\phi}_f\mathbf{A}\right) \tag{16}$$

where, for face $f$ in Figure 1*a*, $\tilde{\phi}_f$ is given by

$$\tilde{\phi}_f = \frac{\phi_{f,0} + \phi_{f,1}}{2} \tag{17}$$

### Boundary Conditions

In addition to cell centers, $\phi$ is also stored at boundary fact centroids. The boundary diffusion flux can then be linearized in the same manner as at an interior face, i.e., using Eq. (12). For the boundary face in Figure 1*b*, this yields

$$D_b = \Gamma_b \frac{(\phi_b - \phi_0)}{db} \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{e}_b} + \Gamma_b \left( \nabla\phi_0 \cdot \mathbf{A} - \nabla\phi_0 \cdot \hat{e}_b \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{e}_b} \right) \tag{18}$$

where $\phi_b$ is the value at the boundary and $\hat{e}_b$ is the vector from the cell centroid to the boundary face centroid.

In the case of Dirichlet boundary conditions, the primary term is included implicitly and the secondary term explicitly. For Neumann boundary conditions the specified flux can be added directly to the control-volume balance; Eq. (18) is employed to compute the boundary value, $\phi_b$, for postprocessing. Other boundary specifications, for example, convection and/or radiation conditions at the wall for the energy equation, are easily accommodated within this framework.

### Discretized Equations

The discretization procedure yields the following linear system of equations for $\phi$ at the cell centers:

$$a_p \phi_p = \sum_{\text{nb}} a_{\text{nb}} \phi_{\text{nb}} + S_p \tag{19}$$

Here the summation is over all the neighbors nb of cell $p$. The source term $S_p$ contains any volumetric sources of $\phi$, the second-order contributions for the convective flux, as well as the secondary diffusion fluxes. Flux contributions at boundaries are also included in $S_p$. Equation (19) is underrelaxed in the manner described in [8].

### Linear Solver

The number of cell neighbors in Eq. (19) is arbitrary for an unstructured mesh. Consequently, familiar line-iterative solvers cannot be used. Instead, the system is solved using an algebraic multigrid procedure similar to that of Hutchinson and Raithby [24].

Hutchinson and Raithby developed their method for structured meshes. They constructed coarse levels by simply removing alternate grid lines in each of the mesh directions $(\xi, \eta)$. On unstructured grids the agglomeration, and consequently, the restriction and prolongation operators, have to be multidimensional.

In the present work, coarse-level cells are constructed by grouping a fine-level cell with $n$ of its neighboring ungrouped cells for which the coefficient $a_{\text{nb}}$ is the largest [25]. Best performance is observed when the group size is 2, leading to a coarse level that is roughly half the size of the fine level. This procedure is applied recursively until no further coarsening is possible. At each level, Gauss-Seidel iteration is used as the relaxation method. A variety of multigrid cycles, such as the V, W, and Brandt cycles [26], have been implemented. For the examples in this article, the V cycle is used for pressure, and the Brandt cycle for all other variables.

The advantage of this algebraic multigrid procedure is that discretization of the governing equations at the coarse levels is avoided. Since the coarsening procedure is based on the coefficients of the discrete equations, grid and flow anisotropies are handled efficiently. Unlike geometric multigrid methods, the coarsening changes appropriately for each linear system, leading to optimum linear solver performance for all transport equations.

### Discretization of the Momentum Equation

The momentum equation is discretized in the same manner as the general scalar equation described above. The source term $F_i$ contains the pressure gradient and components of the stress tensor not included in the standard diffusion term:

$$\frac{\partial}{\partial x_j}\left(\mu\frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\mu\frac{\partial u_l}{\partial x_l} - \delta_{ij}p\right) \tag{20}$$

This is discretized by integrating over the control volume as

$$\sum_f\left(\mu_f\frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\mu_f\frac{\partial u_l}{\partial x_l} - \delta_{ij}p_f\right)A_j \tag{21}$$

The derivatives in Eq. (21) are evaluated at the face by averaging the cell derivatives. The pressure $p_f$ is calculated by averaging the reconstructed face pressure from the two cells neighboring the face.

### Discretization of the Continuity Equation

The discrete continuity equation is written as

$$\sum_f J_f = 0 \tag{22}$$

Since pressure and velocity components are stored at cell centers, computing face mass flow rate by averaging the cell velocity is prone to checkerboarding. To avoid this, a scheme similar to that of Rhie and Chow [27] is used. For the face f in

Figure 1*a*, the mass flow rate is written as

$$J_f = \rho_f \mathbf{A} \cdot \left( \frac{\mathbf{V}_0^* + \mathbf{V}_1^*}{2} \right) - \frac{\rho_f (\Delta \mathscr{V}_0 + \Delta \mathscr{V}_1)}{(\bar{a}_0 + \bar{a}_1)} \left( \frac{p_1 - p_0}{ds} - \overline{\nabla p_f} \cdot \mathbf{e}_s \right) \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{e}_s} \tag{23}$$

where $\mathbf{V}^*$ is the velocity field that satisfies momentum conservation, $\bar{a}$ is the average of the momentum equation $a_p$ coefficients and $\overline{\nabla p_f}$ is the averaged pressure gradient. Using the solution of the underrelaxed momentum equations for $\mathbf{V}^*$ leads to underrelaxation factor dependence in the converged solution; the treatment suggested by Majumdar [28] is used to avoid this problem. Since the pressure gradient term in the momentum equations is weighted with the volume, volume-weighted averaging of $\overline{\nabla p_f}$ is crucial for unstructured meshes, where the cell volumes of adjacent cells can differ greatly.

## Pressure-Correction Equation

The SIMPLE algorithm [8] is used for pressure-velocity coupling. Accordingly, we require that

$$\sum_f (J_f^* + J_f') = 0 \tag{24}$$

where $J_f^*$ is the flow rate computed from velocities satisfying the discrete momentum equations. We define the correction $J_f'$ as

$$J_f' = - \frac{\rho_f (\Delta \mathscr{V}_0 + \Delta \mathscr{V}_1)}{(\bar{a}_0 + \bar{a}_1)} \left( \frac{p_1' - p_0'}{ds} \right) \frac{\mathbf{A} \cdot \mathbf{A}}{\mathbf{A} \cdot \hat{e}_s} \tag{25}$$

Substituting Eq. (25) into Eq. (24) yields the pressure-correction equation:

$$a_p p_p' = \sum_{\mathrm{nb}} a_{\mathrm{nb}} p_{\mathrm{nb}}' + b \tag{26}$$

Here, as in [8], $a_p = \Sigma_{\mathrm{nb}} a_{\mathrm{nb}}$. The term $b$ is the net mass inflow into the cell. The summation is over all cell neighbors of cell $p$.

Once the corrections $p'$ are available, the face mass flow rate, cell pressure, and cell velocity are corrected using

$$J_f = J_f^* + J_f' \tag{27}$$

$$p_p = p_p^* + \alpha_p p_p' \tag{28}$$

$$u_i = u_i^* - \frac{\Sigma_f (A_i p_f')}{a_p^i} \tag{29}$$

Here, $p^*$ and $u_i^*$ are the values prevailing after the solution of the momentum equations, $\alpha_p$ is the underrelaxation factor for pressure, and $a_p^i$ is the center coefficient for the $i$ momentum equation. The face pressure correction $p_f'$ is computed by averaging the $p'$ values of the cell neighbors of face $f$.

### Grid Adaption

The solution algorithm has been carefully formulated to permit both traditional $h$-refinement as well as grids with "hanging nodes" resulting from nonconformal grid adaption. In the latter case, cells marked for adaption are subdivided by introducing midpoint nodes at edges and/or cell and face centers. Since no element-specific shape functions or interpolation formulas are used and all discretization operations are based on faces, the algorithm is applicable to arbitrary polyhedra. Nonconformal interfaces created by hanging-node adaption thus require no special treatment. For example, the undivided cell C1 in Figure 1*a* next to a divided cell is simply considered to be a polyhedron, bounded by the five faces $a$–$b$, $b$–$c$, $c$–$d$, $d$–$e$, and $e$–$a$.

## RESULTS

In this section, we apply the method developed above to a number of benchmark problems in the literature. These tests seek to establish the accuracy and convergence characteristics of the method, and to demonstrate that a variety of mesh topologies can be used. In particular, care is taken to use truly unstructured meshes when making comparisons with structured-mesh benchmarks. This is critical because unstructured meshes created, for example, by triangulating an underlying structured mesh are not good indicators of performance for realistic industrial applications.

### Flow in a Three-Dimensional Cavity

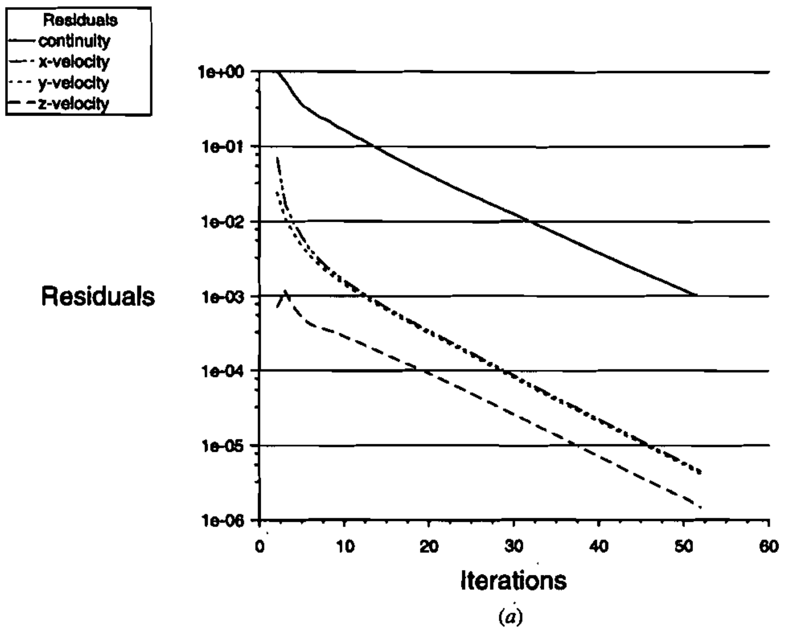
Here we compute flow in a three-dimensional square cavity driven by a moving lid for a Reynolds number of 100, based on the lid velocity and the cavity dimension. We use structured hexahedral as well as unstructured tetrahedral meshes and compare against solutions in the literature. The objective of the computations is to use this popular benchmark to establish that the algorithm is robust, and that the number of iterations to convergence is similar to that of typical segregated solvers.

The solution is considered converged when all scaled residuals fall below $10^{-3}$. For the $x$ and $y$ velocities, the scaled residual is defined as

$$R = \frac{\Sigma_{\text{cells}}(|\Sigma_{\text{nb}} a_{\text{nb}} \phi_{\text{nb}} - b|)}{\Sigma_{\text{cells}} |a_p \phi_p|} \tag{30}$$

For the continuity equation, the scaled residual is defined as

$$R = \frac{\Sigma_{\text{cells}} |\text{mass imbalance}|}{\Sigma_{\text{cells}} |\text{mass imbalance}|_{\text{ref}}} \tag{31}$$



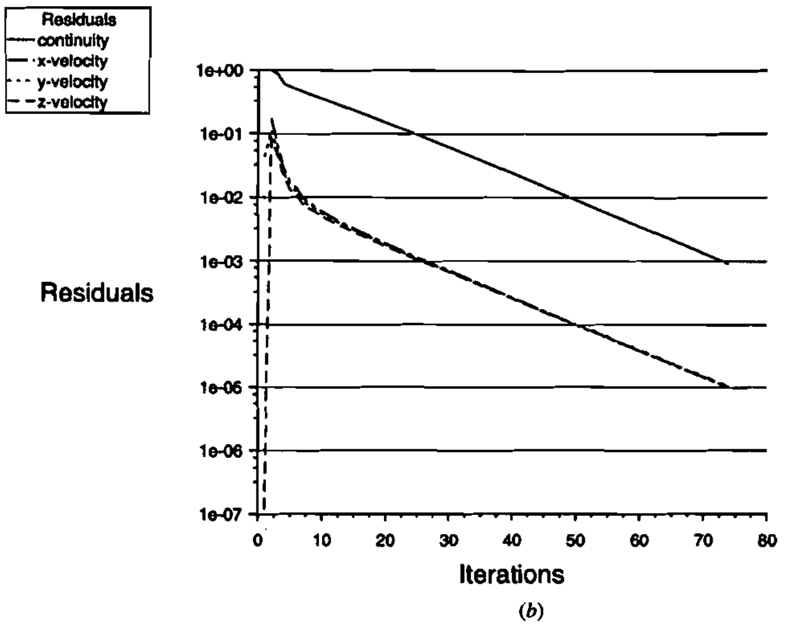
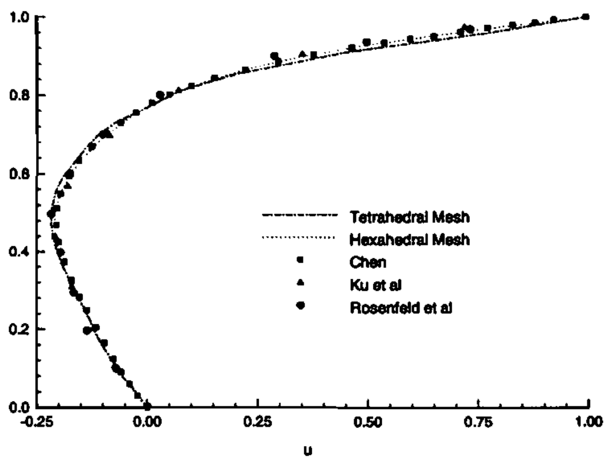


**Figure 3.** 3D Cavity: (*a*) residual history for hexahedral mesh; (*b*) residual history for tetrahedral mesh.



(c)

**Figure 3.** 3D Cavity (*Continued*): (c) $x$ velocity on centerline.

where the mass imbalance is the source term in Eq. (26). For a closed cavity, there is no throughflow to provide a normalizing value, consequently the maximum imbalance during the first five iterations is used as the reference value.

We use two meshes of 19,800 hexahedra and 22,429 tetrahedra, respectively; only one-half of the cavity is solved because of symmetry. Underrelaxation factors for velocity and pressure are 0.95 and 0.1, respectively. Figures 3*a* and 3*b* show that the convergence behavior of the hexahedral and tetrahedral meshes is similar, with the hexahedral case converging in 52 iterations and the tetrahedral case in 73 iterations. The tetrahedral mesh takes somewhat longer because secondary gradients are nonzero on this mesh, whereas the hexahedral mesh is orthogonal. Similar iteration counts have been reported in the literature [29]. The $x$ velocity on the centerline of the cavity is presented in Figure 3*c* along with the results of [30–32]; good agreement is found. It is important to note that the hexahedral case yields a discretization very similar to structured-mesh discretizations presented by Peric [19] and others. The primary difference lies in the implementation of the second-order differencing scheme which, in our case, employs linear reconstruction.

## Flow in a Skewed Lid-Driven Cavity

Demirdzic et al. [33] have published benchmark solutions for lid-driven flow in a skewed cavity, as shown in Figure 4*a*. We consider here the case $\theta = 30°$, at a Reynolds number Re $(= UL/\nu) = 1{,}000$. This case is challenging for structured quadrilateral meshes because of the extreme skewness of the cavity. We solve the problem using both quadrilateral and triangular meshes. The objective of the calculation is (1) to compare computed results with those of [33] and to establish accuracy and convergence to a mesh-independent solution, and (2) to compare the performance of structured quadrilaterals and unstructured triangular cells.

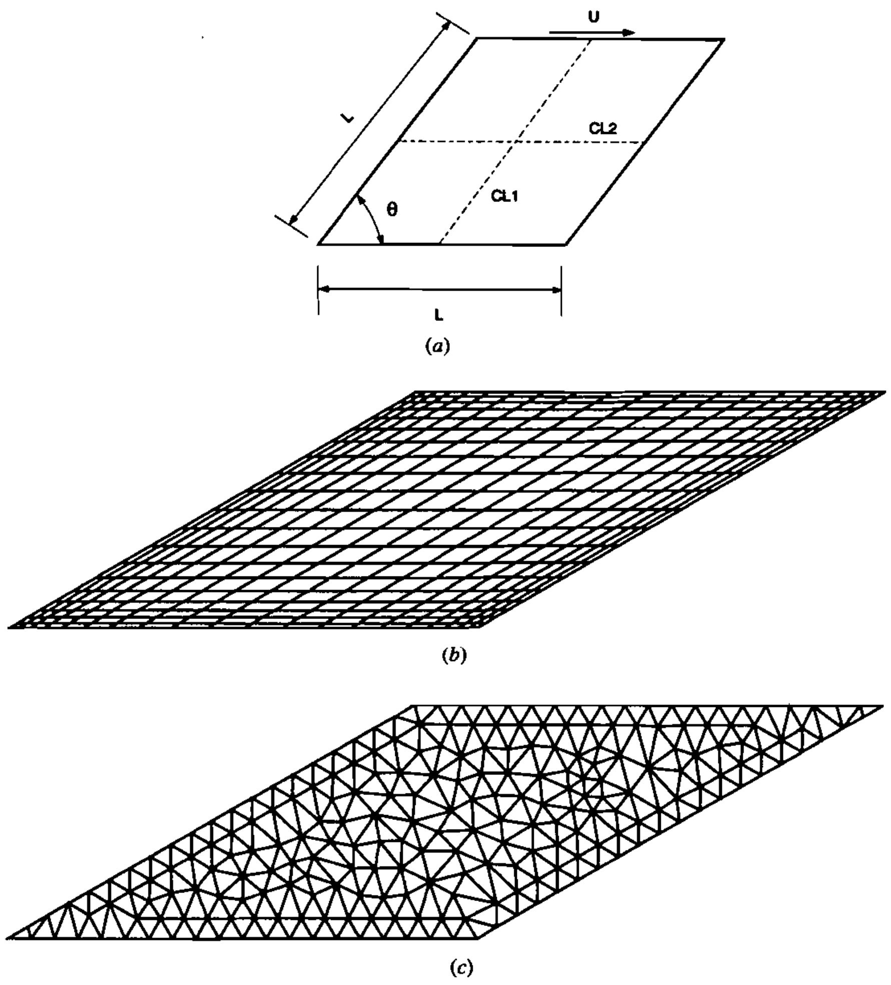


**Figure 4.** Skewed cavity: (*a*) schematic; (*b*) quadrilateral mesh; (*c*) triangular mesh.

To establish the convergence of the scheme to a mesh-independent solution, flow in the cavity is computed using a structured quadrilateral mesh of 20 × 20, 40 × 40, 80 × 80, 160 × 160, and 320 × 320 cells. The 20 × 20 mesh is shown in Figure 4*b*. The finer meshes are created by subdividing each cell into four. The $x$ velocity on the centerline CL1 is shown in Figure 5*a*, along with the tabulated benchmark values from [33] obtained using 320 × 320 quadrilateral cells. The $y$ velocity on the centerline CL2 is shown in Figure 6*a*, along with benchmark values. We find that the solutions for mesh densities greater than 80 × 80 cells are indistinguishable from the benchmark and from each other. This rate of convergence to a mesh-independent solution is similar to that of the second-order structured mesh scheme used in [33].

A similar test of mesh independence is done using unstructured triangular meshes as well. We start with a coarse mesh of 386 triangles, shown in Figure 4*c*. This mesh is approximately equivalent to the 20 × 20 quadrilateral mesh used above. We quadruple the mesh successively by dividing each triangle into four connecting face centroids. Computations are done for 1,544, 6,176, and 24,704 cells, respectively. The *x* velocity on the centerline CL1 is presented in Figure 5*b*, and the *y* velocity on CL2 in Figure 6*b*. Mesh-independent solutions are obtained for mesh densities greater than 6,176 cells, similar to quadrilaterals. The rate of convergence to a mesh-independent solution is also similar to that observed for quadrilaterals.

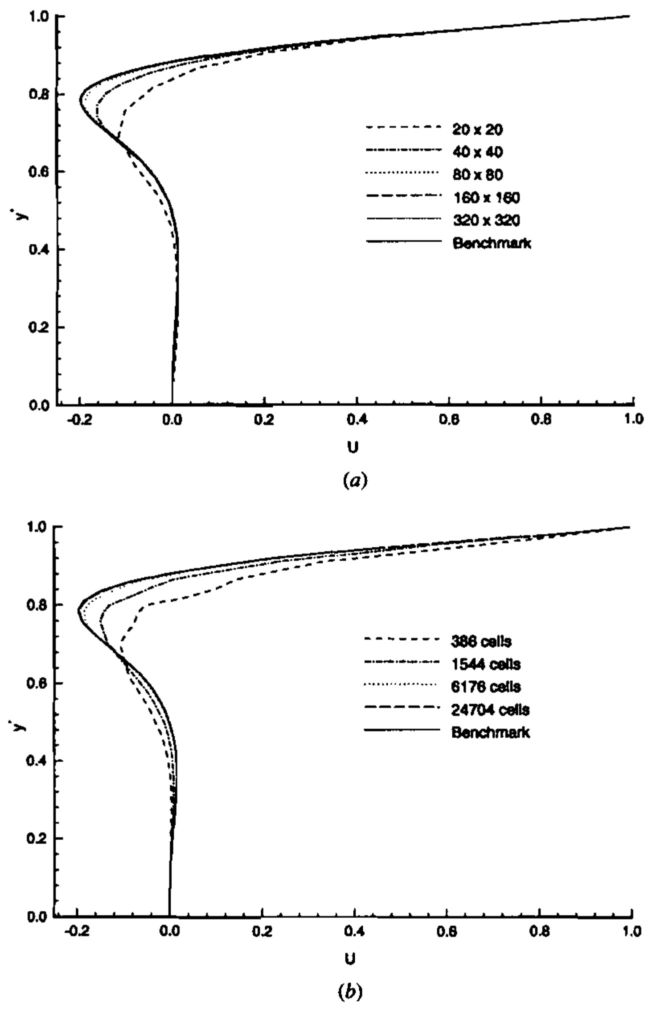
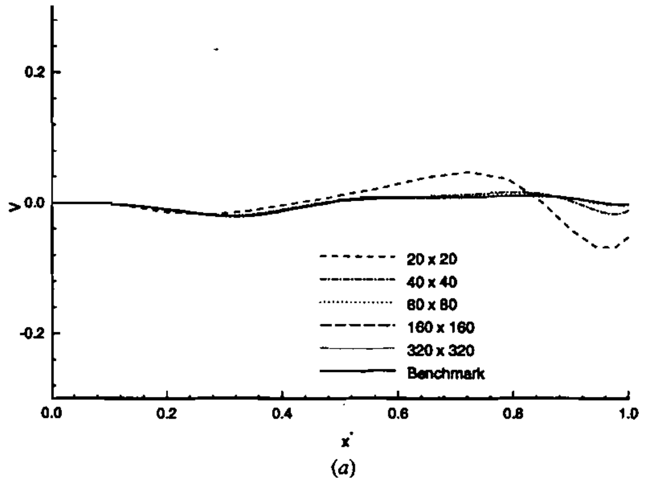


Figure 5. Skewed cavity, *x* velocity on CL1: (*a*) quadrilateral mesh; (*b*) triangular mesh.



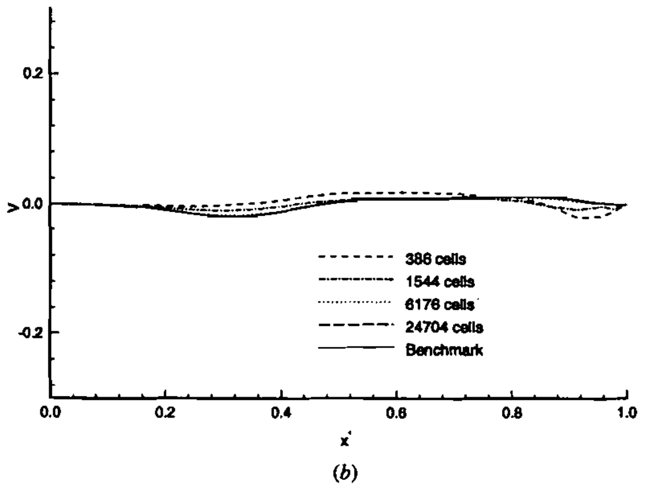


**Figure 6.** Skewed cavity, $y$ velocity on CL2: (*a*) quadrilateral mesh; (*b*) triangular mesh.

## Natural Convection in a Cavity

Here we compute flow and heat transfer due to a heated cylinder in a square cavity, as shown in Figure 7. The cylinder center is displaced from the center of the cavity by $\delta$. The Rayleigh number based on the cylinder diameter is $10^6$, and the Prandtl number is 0.1. $\delta/L = 0.1$ and $d/L = 0.4$ are used. The Boussinesq approximation is used to model buoyancy. A benchmark solution for this configuration has been published by Demirdzic et al. [33] using a structured mesh of $256 \times 128$ cells. The objective of this computation is to demonstrate the use of hybrid meshes and hanging-node adaption, and to establish that the results so computed match published data.

Quadrilaterals are used around the cylinder and along the cavity walls; triangles are used elsewhere. The mesh is adapted to velocity magnitude using

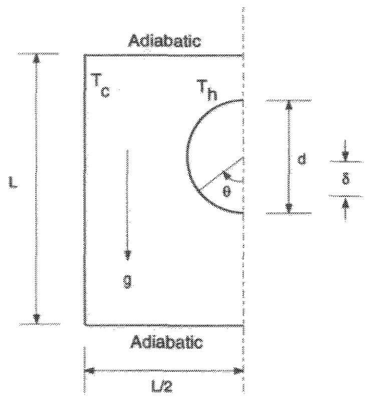


**Figure 7.** Natural convection in a cavity—schematic.

hanging-node adaption. The final mesh is shown in the left half of Figure 8 and contains 16,113 cells. It captures the hot plume rising around the cylinder, and the downward flowing stream at the cold wall as seen in the right half of the figure.

Plots of the local Nusselt number {defined as $\mathrm{Nu} = q''L/[k(T_h - T_c)]$, where $q''$ is the local heat flux} along the cold wall and along the hot cylinder are shown in Figure 9. The results match those of [33] well, even though the mesh is only half as dense.

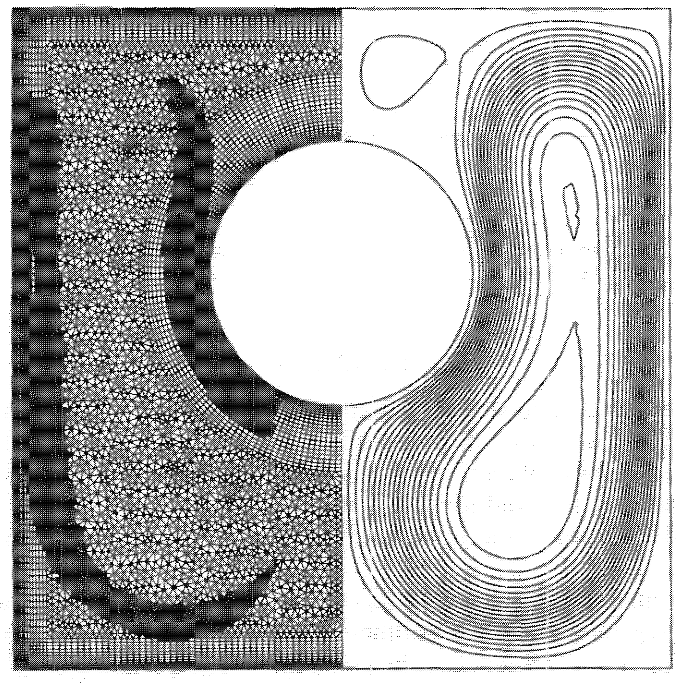
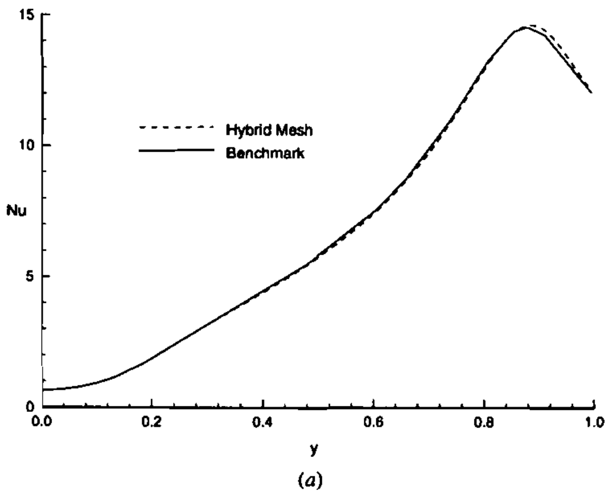


**Figure 8.** Natural convection in a cavity—adapted hybrid mesh and streamlines.



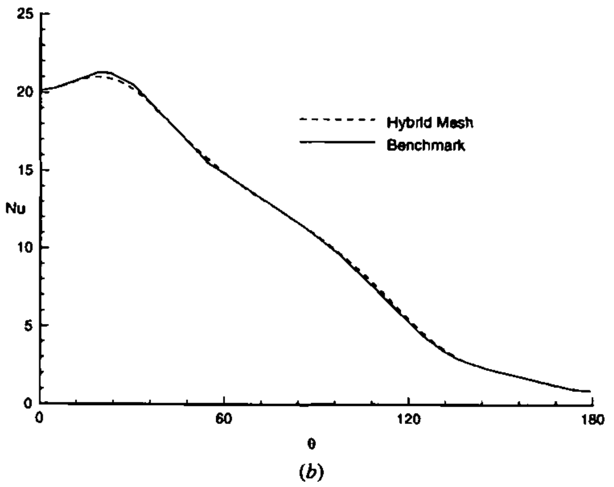


**Figure 9.** Natural convection in a cavity—Nusselt number: (*a*) cold wall; (*b*) hot wall.

## CLOSURE

A conservative finite-volume scheme has been developed for computing flow and heat transfer on unstructured meshes. The method admits arbitrary convex polyhedra, and allows solution adaptivity. Care is taken in the discretization and solution procedures to avoid formulations that are cell-shape-specific. The resulting scheme demonstrates the same accuracy and robustness as typical structured mesh schemes.

The formulation presented in this article is second-order-accurate and employs the segregated SIMPLE algorithm for pressure–velocity coupling. An interesting area for future research is the development of pressure-based schemes of arbitrarily high order using reconstruction ideas. Another area is the development of point-coupled and segregated multigrid schemes [34, 35] in the unstructured-mesh context. Parallel processing for pressure-based, implicit unstructured mesh schemes is also a challenging area. With these and other enhancements, and the development of physical models for unstructured meshes, useful predictive tools may be developed for practical industrial applications.

## REFERENCES

1. A. Jameson, T. J. Baker, and N. P. Weatherhill, Calculation of Inviscid Flow over a Complete Aircraft, AIAA-86-0103, 1986.
2. D. Mavriplis, A. Jameson, and L. Martinelli, Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes, AIAA-89-0120, 1989.
3. A. J. Chorin, A Numerical Method for Solving Incompressible Viscous Flow Problems, *J. Comput. Phys.*, vol. 2, pp. 12–26, 1967.
4. D. Kwak, J. L. Chang, S. P. Shanks, and S. R. Chakravarthy, A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables, *AIAA J.*, vol. 23, pp. 390–396, 1985.
5. J. M. Weiss and W. A. Smith, Solution of Unsteady, Low Mach Number Flow Using a Preconditioned, Multi-Stage Scheme on an Unstructured Mesh, AIAA-93-3392, 1993.
6. J. M. Weiss and W. A. Smith, Preconditioning Applied to Variable and Constant Density Time-Accurate Flows on Unstructured Meshes, AIAA-94-2209, 1994.
7. V. Venkatakrishnan and D. Mavriplis, Implicit Solvers for Unstructured Meshes, *AIAA 10th CFD Conf. Proc.*, pp. 115–124, 1991.
8. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 198C.
9. P. Hood and C. Taylor, Navier-Stokes Equations Using Mixed Interpolations, in J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher, and C. Taylor (eds.), *Finite Element Methods in Flow Problems*, pp. 121–132, UAH Press, Huntsville, AL, 1974.
10. P. S. Huyakorn, C. Taylor, R. L. Lee, and P. M. Gresho, A Comparison of Various Mixed-Interpolation Finite Elements in the Velocity-Pressure Formulation of the Navier-Stokes Equations, *Computers and Fluids*, vol. 6, pp. 25–35, 1978.
11. T. E. Tezduyar, R. Shih, S. Mittal, and S. E. Ray, Incompressible Flow Computations with Stabilized Bilinear and Linear Equal-Order Interpolation Velocity-Pressure Elements, University of Minnesota Supercomputer Institute Research Report UMSI90/165, 1990.
12. A. N. Brooks and T. J. R. Hughes, Streamline Upwind/Petrov-Galerkin Formulations for Convection-Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations, *Comput. Meth. Appl. Mech. Eng.*, vol. 32, pp. 199–259, 1982.
13. B. R. Baliga and S. V. Patankar, A Control-Volume Finite Element Method for Two-Dimensional Fluid Flow and Heat Transfer, *Numer. Heat Transfer*, vol. 6, pp. 245–261, 1983.
14. G. E. Schneider and M. J. Raw, Control-Volume Finite-Element Method for Heat Transfer and Fluid Flow Using Co-Located Variables—1. Computational Procedures, *Numer. Heat Transfer*, vol. 11, pp. 363–390, 1987.
15. C. Prakash and S. V. Patankar, A Control-Volume Based Finite Element Method for Solving the Navier-Stokes Equations Using Equal-Order Velocity-Pressure Interpolation, *Numer. Heat Transfer*, vol. 8, pp. 259–280, 1985.
16. Y. Jiang and A. J. Przekwas, Implicit, Pressure-Based Incompressible Navier-Stokes Equations Solver for Unstructured Meshes, AIAA-94-0305, 1994.
17. I. Demirdzic and S. Muzaferija, Numerical Method for Coupled Fluid Flow, Heat Transfer and Stress Analysis Using Unstructured Moving Meshes with Cells of Arbitrary Topology, *Comput. Meth. Appl. Mech. Eng.*, vol. 125, pp. 235–255, 1995.

18. L. Davidson, A Pressure Correction Method for Unstructured Meshes with Arbitrary Control Volumes, *Int. J. Numer. Meth. Fluids*, vol. 22, pp. 265–281, 1996.
19. M. Peric, A Finite Volume Method for the Prediction of Three-Dimensional Fluid Flow in Complex Ducts, Ph.D. Thesis, University of London, UK, 1985.
20. K. C. Karki and S. V. Patankar, Pressure Based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations, *AIAA J.*, vol. 27, pp. 1167–1174, 1989.
21. M. Thomadakis and M. Leschziner, Numerical Simulation of Viscous Incompressible Flows Using a Pressure-Correction Method and Unstructured Grids, in *Computational Fluid Dynamics 94*, pp. 325–332, John Wiley and Sons, Chichester, UK, 1994.
22. T. J. Barth, 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, AGARD Report 787, 1992.
23. V. Venkatakrishnan, On the Accuracy of Limiters and Convergence to Steady State Solutions, AIAA 93-0880, Jan 1993.
24. B. R. Hutchinson and G. D. Raithby, A Multigrid Method Based on the Additive Correction Strategy, *Numer. Heat Transfer*, vol. 9, pp. 511–537, 1986.
25. R. D. Lonsdale, An Algebraic Multigrid Scheme for Solving the Navier-Stokes Equations on Unstructured Meshes, *Proc. 7th Int. Conf. on Numerical Methods in Laminar and Turbulent Flow*, pp. 1432–1442, Stanford University, Palo Alto, CA, 1991.
26. A. Brandt, Multi-Level Adaptive Solutions to Boundary Value Problems, *Math. Comput.*, vol. 31, pp. 333–390, 1977.
27. C. M. Rhie and W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA J.*, vol. 21, pp. 1523–1532, 1983.
28. S. Majumdar, Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Nonstaggered Grids, *Numer. Heat Transfer*, vol. 13, pp. 125–132, 1988.
29. K. H. Chen and R. H. Pletcher, Primitive Variable, Strongly Implicit Calculation Procedure for Viscous Flows at All Speeds, *AIAA J.*, vol. 29, 1241–1249, 1991.
30. K. H. Chen, A Primitive Variable, Strongly Implicit Calculation Procedure for Two and Three-Dimensional Unsteady Viscous Flows: Applications to Compressible and Incompressible Flows Including Flows with Free Surfaces, Ph.D. thesis, Iowa State University, Ames, IA, 1990.
31. H. Ku, R. Hirsch, and T. Taylor, A Pseudospectral Method for Solution of Three Dimensional Navier-Stokes Equations, *J. Comput. Phys.*, vol. 70, pp. 439–462, 1987.
32. M. Rosenfeld, D. Kwak, and M. Vinokur, A Solution Method for the Unsteady and Incompressible Navier-Stokes Equations in Generalized Coordinate Systems, AIAA-88-0718, 1988.
33. I. Demirdzic, Z. Lilek, and M. Peric, Fluid Flow and Heat Transfer Test Problems for Non-Orthogonal Grids: Bench-Mark Solutions, *Int. J. Numer. Meth. Fluids*, vol. 15, pp. 329–354, 1992.
34. R. Jyotsna and S. P. Vanka, Multigrid Calculation of Steady, Viscous Flow in a Triangular Cavity, *J. Comput. Phys.*, vol. 122, pp. 107–117, 1995.
35. R. Webster, An Algebraic Multigrid Solver for Navier-Stokes Problems, *Int. J. Numer. Meth. Fluids*, vol. 18, pp. 761–780, 1994.

## APPENDIX

Consider the face $f$ in Figure 2. The $\xi$ direction is aligned with the vector joining cell centroids; the $\eta$ direction lies in the direction joining the vertices of the face. We may write

$$\nabla\phi \cdot \mathbf{A} = \phi_\xi(\xi_x A_x + \xi_y A_y) + \phi_\eta(\eta_x A_x + \eta_y A_y) \tag{32}$$

Expressing the transformation metrics in terms of derivatives of $(x, y)$, we have

$$\nabla\phi \cdot \mathbf{A} = \phi_\xi\left(\frac{y_\eta A_x - x_\eta A_y}{x_\xi y_\eta - x_\eta y_\xi}\right) + \phi_\eta\left(\frac{-y_\xi A_x - x_\xi A_y}{x_\xi y_\eta - x_\eta y_\xi}\right) \tag{33}$$

For the structured body-fitted grid shown in Figure 2, writing consistent approximations for the derivatives, we have the following expressions:

$$\begin{aligned}
\phi_\xi &= \phi_1 - \phi_0 \\
x_\xi &= x_1 - x_0 \\
y_\xi &= y_1 - y_0 \\
\phi_\eta &= \phi_b - \phi_a \\
x_\eta &= x_b - x_a \\
y_\eta &= y_b - y_a
\end{aligned}$$

Substituting in Eq. (33), we obtain

$$\begin{aligned}
\nabla\phi \cdot \mathbf{A} = &\frac{\phi_1 - \phi_0}{(x_1 - x_0)(y_b - y_a) - (x_b - x_a)(y_1 - y_0)}\left[(y_b - y_a)A_x - (x_b - x_a)A_y\right] \\
&+ \frac{(\phi_b - \phi_a)}{(x_1 - x_0)(y_b - y_a) - (x_b - x_a)(y_1 - y_0)} \\
&\times\left[-(y_1 - y_0)A_x + (x_1 - x_0)A_y\right]
\end{aligned} \tag{34}$$

From the geometry in Figure 2 it is clear that

$$A_x = (y_b - y_a) \tag{35}$$

$$A_y = -(x_b - x_a) \tag{36}$$

We also note that the unit vector in the direction joining the two cell centroids is

$$\hat{e}_s = \frac{(x_1 - x_0)\hat{\imath} + (y_1 - y_0)\hat{\jmath}}{ds} \tag{37}$$

where $ds$ is the distance between the centroids. The tangent vector along the face has the components

$$\hat{e}_t = \frac{(x_b - x_a)\hat{\imath} + (y_b - y_a)\hat{\jmath}}{A} \tag{38}$$

Using these relations, we may write the diffusion flux on face $f$ as

$$\nabla\phi \cdot \mathbf{A} = \frac{\phi_1 - \phi_0}{ds}\frac{\mathbf{A}\cdot\mathbf{A}}{\mathbf{A}\cdot\hat{e}_s} + \frac{\phi_b - \phi_a}{A}\frac{\mathbf{A}\cdot\mathbf{A}}{\mathbf{A}\cdot\hat{e}_s}\hat{e}_t\cdot\hat{e}_s \tag{39}$$