JOURNAL OF COMPUTATIONAL PHYSICS **138,** 766–787 (1997)
ARTICLE NO. CP975853

Finite-Volume CFD Procedure and Adaptive Error Control Strategy for Grids of Arbitrary Topology

Samir Muzaferija[1] and David Gosman

Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London, SW7 2BX, England

Received November 7, 1996

This paper outlines the development and application of a solution-adaptive local grid refinement procedure for numerical fluid dynamic calculations in complex domains involving body-fitted unstructured meshes. A new space discretization practice and an error estimation technique were developed to facilitate adaptive space discretization (*h*-refinement) using cells of arbitrary topology. The methodology enables implicit, consistent, and uniform treatment throughout the entire computational domain including the interface between refined regions and the rest of the computational mesh. It is demonstrated on a number of test cases involving both laminar and turbulent flows, in which initially regular hexahedral meshes are refined by cell subdivision. Encouraging results are obtained. © 1997 Academic Press

Key Words: finite volume; polyhedral discretization elements; error detection; local grid refinement.

1. INTRODUCTION

For a long time the search for solutions free from numerical error has been and will be the ultimate goal of CFD research. An indisputable way to reduce the numerical error is by reducing the size of the discretization elements, so-called "*h*-refinement." For efficient and flexible grid generation and error-controlled local grid refinement, it is a significant advantage to be able to utilize discretization elements of arbitrary topology. This is particularly obvious in CFD simulations in

[1] Current address: Universität Hamburg, Institut für Schiffbau, Lämmersieth 90, D-22305 Hamburg, Germany.

complex geometrical configurations, where accurate description of boundaries is desired and local grid refinement is to be carried out only in the regions where it is necessary, without disturbing the rest of the computational mesh.

The information about the regions where the grid should be refined/coarsened ideally should be provided by an error estimator. Error estimation for fluid flow calculations is not an easy task. The Navier–Stokes equations, together with the transport equations for turbulence modeling quantities, are a coupled, nonlinear system, and errors present in any one of these fields in general will affect the solution of all others, in a nonlinear manner difficult to describe accurately.

The discretization error describes the deviation from the analytical solution of the set of differential equations, but it is not directly accessible. However, it is possible to construct approximations to this error. There are a number of adaptive methods in computational fluid mechanics which are designed to be used in combination with finite-element and finite-difference schemes [1–3]. Adaptive finite volume (FV) discretization and error estimation techniques presented in the literature are less numerous. Error estimation for the FV method was originally examined in conjunction with turbulence modeling [4]. The use of upwind differencing introduces excessive amounts of numerical diffusion which interferes with the turbulent diffusion introduced by the turbulence model. The numerical diffusion is estimated by comparing the convection flux estimations based on upwind and central differencing [4, 5]. In a later work by Tattersall and McGuirk [6], the numerical diffusion estimate has been coupled with an adaptive node-movement technique.

The cell-to-cell imbalances in angular momentum and kinetic energy are proposed by Haworth *et al.* [7] to characterize the local solution error. This method has been tested on a transient flow problem in an internal combustion engine. The method is not capable of estimating the absolute error levels or the error present in the transport of scalar quantites (e.g., thermal energy or turbulence kinetic energy).

Richardson extrapolation is the most popular error estimation method in FV calculations. It has been used extensively on a variety of situations, ranging from supersonic flows [8] to incompressible problems [9, 10]. Richardson extrapolation is quite reliable on fine grids, since it takes into account not only the smoothness of the spatial variation of the dependent variables but also the nonlinearities and interequation couplings. The method naturally couples with the use of multigrid acceleration techniques, where solutions on grids with different cell size are already available. However, in order to obtain an error estimate using Richardson extrapolation, it is necessary to obtain solutions on at least two grids of spacing differing by a factor (typically two). A uniformly refined three-dimensional hexahedral grid has eight times more control volumes (CVs) than a coarse one. For problems where the geometry and/or the physics to be modeled are complex, the coarsest mesh on which one can perform sensible calculations already has a very large number of CVs and a uniform refinement throughout the computational domain, just to estimate the error, often cannot be afforded.

In the framework of *h*-refinement for FV methods, several different ways of mesh refinement have been suggested, with different implications with respect to the speed, stability, accuracy, and complexity of the flow solver. A computational grid made of a sequence of overlapping patches of increased fineness is used by Caruso [11]. Each of the overlapping patches is an orthogonal and structured grid. In the flow solver, each "patch" is treated independently, with the information transfer between the different parts of the mesh performed through the "patch" boundary conditions. A sensitive point of the method is the transfer of information between the overlapping grids via internal boundary conditions. This is done explicitly, resulting in weaker coupling and slower convergence. Also, resolution problems have been reported at places where flow features intersect with "patch" boundaries [8]. A clustering algorithm is used by Berger and Oliger [12] to optimize the construction of patches, their position, and mesh size. It uses concepts from pattern recognition and artificial intelligence theory. Thompson and Ferziger [10] used the same algorithm in the simulation of viscous flows. Tu and Fuchs [13] investigated composite or *Chimera* grids and multigrid methods in calculation of unsteady flows in IC engines. Perng and Street [14] introduced a new method to obtain multiple domain solutions for incompressible flows, which updates the velocity field independently on each subgrid and solves the pressure field globally by sweeping through the subgrids. This procedure reduces the cost of computation significantly compared with conventional methods which iterate both the momentum and the pressure equations through subdomains. The new method also required that the grid points from different subgrids in the overlapping zone be coincident.

A refinement procedure in which the refinement patches are embedded into the original mesh, thus removing the interpolation problems, is suggested by Chen *et al.* [15]. The resulting mesh is then treated in a multiblock manner. Although this approach presents a considerable improvement over earlier work, it is not appropriate for situations requiring large number of embedded refinement levels, as the number of blocks becomes so large that it significantly impairs the performance of the code, as reported in [15]. Clustering of grid points is also necessary in order to obtain a grid made of a number of structured blocks.

Tetrahedral grids offer geometrical flexibility and allow simple and highly localized refinement. Although very good results are produced in inviscid calculations, the extension of the method to viscous and turbulent flows has been somewhat less successful. Vilsmeier and Hänel [16] have developed an adaptive FV algorithm on tetrahedral meshes for Euler and Navier–Stokes equations using *h*-refinement on a cell-by-cell basis. Virtual stretching of triangular elements has been introduced to provide the capability of mesh alignment. It is performed in the vicinity of walls and in regions of high shear. Unfortunately, this results in high distortion of the mesh, decreasing the accuracy of the method.

In this paper we present a novel FV discretization practice that does not rely on any specific cell topology, together with a local error estimation technique that is very natural for FV discretization, is simple to implement, and is based on a Taylor series expansion analysis on a given grid. We also present a solution procedure for the discretized equations which is a variant of the well-known pressure-based SIMPLE algorithm [17], originally developed for incompressible flows but later extended to allow for compressibility [18]. The present applications are to incompressible cases.

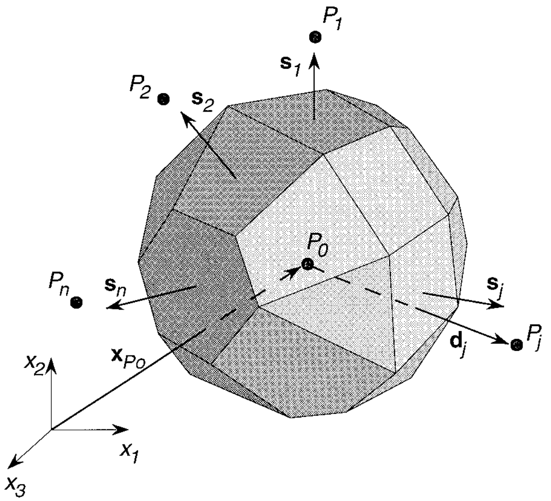

FIG. 1. A general polyhedral control volume and the notation used.

The next section describes the space and equation discretization adopted by the authors. The local error estimation method and refinement practices are then described in Section 3. Section 4 presents several application examples.

2. DISCRETIZATION PROCEDURE

The integral forms of the time-averaged conservation laws for mass, linear momentum, turbulent kinetic energy, and turbulent kinetic energy dissipation rate are used to determine the behavior of the fluid flow. All conservation equations have the same general form, that for an arbitrary spatial region of volume V bounded by closed surface S can be written as

$$\int_S (\rho\mathbf{v}\Phi - \lambda_\Phi \nabla\Phi) \cdot \mathrm{d}\mathbf{s} = \int_V Q_\Phi \, \mathrm{d}V, \tag{1}$$

where ρ is the fluid density, $\mathbf{v}$ is the velocity vector, Φ stands for any conserved quantity, λ_Φ is the associated diffusion coefficient, $\mathrm{d}\mathbf{s}$ is the outward-pointing surface-element vector, and Q_Φ is the volumetric source of Φ.

Meshes made of polyhedral cells bounded by cell faces $\mathbf{s}_j$ (see Fig. 1), and in general of different topology, combined in an arbitrary way, are allowed for the space discretization in the present study. The computational points are located in the centers of the CVs (“the cell-centered” arrangement), and all variables share the same CV (“the colocated” or “nonstaggered” arrangement).

The assumption of linear spatial variations of the dependent variables and the midpoint rule approximation of the surface and volume integrals in Eq. (1) are used for equation discretization, leading to a second-order scheme. The unknown coefficients of a linear profile may be determined by using the values of dependent variables of the nearest neighbors of cell P_0 only, since for the most simple cell topology (tetrahedral), the number of nearest neighbors (four) is sufficient to determine four unknown coefficients. The fact that only nearest neighbors are used results in a compact computational molecule. The assumption about linear variation of Φ within the CV can be written as

$$\phi(x) = \phi_{P_0} + (\nabla\phi)_{P_0} \cdot (\mathbf{x} - \mathbf{x}_{P_0}), \tag{2}$$

where $\mathbf{x}_{P_0}$ is the position vector of point P_0 and $(\nabla\phi)_{P_0}$ is an approximation of the gradient of ϕ at point P_0. The lowercase ϕ in Eq. (2) is introduced to indicate that ϕ is an approximation of Φ and that in a general case these two are different. The unknown three components of $(\nabla\phi)_{P_0}$ are to be determined by demanding that the profile (2) fits the values of ϕ at chosen locations. An attempt to fit the values of ϕ at all nearest neighbors results in an overdetermined algebraic system. In the present work a least-squares fit of Eq. (2) to the set of the nearest neighbor values is proposed to calculate $(\nabla\phi)_{P_0}$. The result can be written in matrix form as

$$(\nabla\phi)_{P_0} = \mathbf{G}^{-1}\mathbf{h}. \tag{3}$$

The components h_k of the vector $\mathbf{h}$ and the coefficients g_{kl} of the 3×3 matrix $\mathbf{G}$ are defined by

$$\begin{aligned} h_k &= \sum_{j=1}^{n} (\phi_{P_j} - \phi_{P_0}) d_j^k, \\ g_{kl} &= \sum_{j=1}^{n} d_j^k\, d_j^l, \end{aligned} \tag{4}$$

where n is the number of neighbor cells that share cell faces with cell P_0, and d_j^k is the kth Cartesian component of the vector $\mathbf{d}_j$ (Fig. 1). It is noteworthy that the matrix $\mathbf{G}$ is symmetric, that its coefficients depend on the cell geometry only, and that it is therefore the same for all dependent variables. If there is no mesh motion, it is enough to calculate $\mathbf{G}^{-1}$ once and to store its six independent coefficients for each CV.

The value of dependent variable at the cell face ϕ_j that features in the convective flux C_j is estimated using the expression

$$\phi_j = \begin{cases} \phi_{P_0} + [\gamma\nabla\phi_j \cdot (\mathbf{x}_j - \mathbf{x}_{P_0})] & \text{if } F_j \geq 0, \\ \phi_{P_j} + [\gamma\nabla\phi_j \cdot (\mathbf{x}_j - \mathbf{x}_{P_j})] & \text{if } F_j < 0, \end{cases} \tag{5}$$

where γ is a blending factor; $\nabla\phi_j$ is the gradient at the cell face obtained by interpolating gradients (3) of cell P_0 and P_j; $\mathbf{x}_j$ is the position vector of the center of the cell face j; and F_j is the mass flux through the cell face j. In the $\gamma = 1$ limit the scheme is second-order centered (CD) and it reduces to the first-order upwind (UD) scheme for $\gamma = 0$. A predetermined blend of the CD and the UD schemes (i.e., $0 \leq \gamma \leq 1$) is used in the present study, in the deferred-correction manner

[19]. In this, the UD values are treated implicitly and the differences between first- and second-order approximations (expression in [ ] in (5)) taken at the previous outer iteration level are used explicitly.

The diffusion flux D_j through the cell face j is a function of the gradient of variable Φ at the face. The gradient of the transported variable obtained by (3) is second-order space centered and as such cannot sense oscillations in the solution with period equal to twice that of the characteristic mesh size h. This can be corrected by the explicit addition of recoupling terms [20, 21] achieved in the present study by adding a certain amount of the third-order numerical smoothing, i.e.,

$$\nabla\phi_j^* = \nabla\phi_j + \left(\frac{\phi_{P_j} - \phi_{P_0}}{|\mathbf{d}_j|} - \nabla\phi_j\cdot\frac{\mathbf{d}_j}{|\mathbf{d}_j|}\right)\frac{\mathbf{d}_j}{|\mathbf{d}_j|}. \tag{6}$$

Assembling the contributions from convective and diffusive fluxes through all cell faces and integrating the source term over all N control volumes in the computational domain results in a system of N algebraic equations of the form

$$a_0\phi_{P_0} - \sum_{j=1}^{n} a_j\phi_{P_j} = b_\phi, \tag{7}$$

for each conserved property Φ. The coefficients a_j and source term b_ϕ are defined as

$$a_j = \frac{\lambda_{\phi_j}}{|\mathbf{d}_j|^2}(\mathbf{d}_j\cdot\mathbf{s}_j) - \min(F_j, 0), \tag{8}$$

$$a_0 = \sum_{j=1}^{\mathrm{n}} a_j - a_{Q_\phi}V_{P_0}, \tag{9}$$

$$\begin{aligned} b_\phi = {} & \sum_{j=1}^{n}\lambda_{\phi_j}\left[\nabla\phi_j - \left(\nabla\phi_j\cdot\frac{\mathbf{d}_j}{|\mathbf{d}_j|}\right)\frac{\mathbf{d}_j}{|\mathbf{d}_j|}\right]\cdot\mathbf{s}_j \\ & - \sum_{j=1}^{n}\gamma\nabla\phi_j\cdot[(\mathbf{x}_j - \mathbf{x}_{P_0})\max(F_j, 0) + (\mathbf{x}_j - \mathbf{x}_{P_j})\max(-F_j, 0)] + b_{Q_\phi}V_{P_0}, \end{aligned} \tag{10}$$

where a_{Q_ϕ} and b_{Q_ϕ} are the coefficients of the linearized source term Q_Φ [17].

The pressure does not feature explicitly in the continuity equation, which consequently cannot be considered as "an equation for pressure" and the continuity equation comes just as an additional constraint on the velocity field. This constraint can be satisfied only by adjusting the pressure field. However, pressure is not a conserved property and has no governing transport equation, so it is not immediately clear how this adjustment of pressure is to be performed. This problem is especially pronounced in case of incompressible flows. At the same time, the pressure source term in the momentum equation is calculated using the second-order space-centered scheme. As mentioned earlier, such a scheme can produce correct pressure-gradient field, although the underlying pressure field possessed an unphysical oscillatory profile [17]. The simple and yet efficient way of getting around both of these aforementioned problems follows the idea used by Rhie and Chow [21]. The calculation of the cell face mass flux is arranged to depend not only on the velocity field but also on the pressure field. This influence of the pressure field is expressed in the form of a third-order pressure coupling term, such that mass flux through the cell face j becomes

$$F_j = \rho_j \left\{ \mathbf{v}_j + K_j \left[\frac{1}{2} (\nabla p_{P_0} + \nabla p_{P_j}) \cdot \frac{\mathbf{d}_j}{|\mathbf{d}_j|} - \frac{p_{P_j} - p_{P_0}}{|\mathbf{d}_j|} \right] \frac{\mathbf{s}_j}{|\mathbf{s}_j|} \right\} \cdot \mathbf{s}_j, \tag{11}$$

where $\mathbf{v}_j$ is the velocity vector at the cell face, estimated using the CD scheme, p pressure and K_j is defined as

$$K_j = \frac{1}{2} \left[\left(\frac{V}{a_0^v} \right)_{P_0} + \left(\frac{V}{a_0^v} \right)_{P_j} \right], \tag{12}$$

where a_0^v is the central coefficient (9) of the momentum equation. The pressure coupling term introduces the pressure into the continuity equation in a rather elegant way, such that it can be regarded as an equation for pressure and the SIMPLE algorithm [17] can be used in its standard form. The mass imbalance

$$Q_m = \sum_{j=1}^{n} F_j \tag{13}$$

is to be annihilated by the mass-flux corrections F_j', which are related to the pressure corrections p',

$$Q_m = \sum_{j=1}^{n} F_j' = \sum_{j=1}^{n} a_j^p (p_{P_j}' - p_{P_0}'), \tag{14}$$

where coefficients a_j^p follow from Eq. (11) and are given by

$$a_j^p = -\rho_j K_j \frac{|\mathbf{s}_j|}{|\mathbf{d}_j|}. \tag{15}$$

This formulation of the pressure-correction equation can be used for incompressible and compressible low Mach number flows (Ma < 0.3).

The coupled system of nonlinear algebraic equations which results from the discretization presented above, is linearized using an iterative procedure based on the SIMPLE algorithm. The linearized system of algebraic equations is sparse, and as can be seen from Eq. (8), each nearest neighbor contributes a coefficient to the coefficient matrix. Since the number of cell nearest neighbors is arbitrary, the resulting matrices do not have any specific structure, or in other words the number of nonzero coefficients and their positions in each matrix row are arbitrary. The iterative solvers of systems of linear algebraic equations which utilize any specific matrix structure cannot be used in this case. Fortunately, very efficient conjugate gradient-based solvers [22, 23] do not need any particular matrix configuration in order to be applied and because of this are used in a combination with the discretization presented.

An illustration of the ability of the present method to handle meshes of arbitrary topology is shown in Fig. 6. The application is to lid-driven laminar cavity flow [25], here computed with various shapes of cells, with numbers of faces ranging from 3 to 10. Further results for this case using adaptive refinement will be shown later.

3. ADAPTIVE TECHNIQUE

The assumed profiles do not in general represent the true spatial (and in transient problems temporal) variations of the dependent variable Φ. Consequently the algebraic system based on these assumptions is satisfied by a solution vector ϕ that has a discretization error e, defined as the difference between the exact fields of dependent variables and the discrete solution field ($e = \Phi - \phi$). The numerical accuracy can be improved by reducing the error in the profile assumptions. There are essentially two means by which this can be achieved. One is to alter the spacing of the computational points, either by changing the number (h-method) or by altering their distribution (r-method). The other alternative is to improve the accuracy of the discretization scheme. For a required solution accuracy it can be more economical to solve a higher-order discretization scheme on a coarse grid than a lower-order one on a finer grid, provided the solution is “sufficiently smooth.” However, the higher-order formulae are not likely to be significantly more accurate than lower-order ones if the exact solution contains discontinuities or if the numerical grid is too coarse. An indisputable way for improving the numerical accuracy is grid refinement, in which the distance over which the profile assumptions apply is made smaller, and the details of the assumptions become less important.

The problem with mesh movement (r-method) is that for a given number of grid points, there is no guarantee of obtaining a sufficiently accurate solution. In addition, procedures for redistributing the grid points can produce distorted CVs. Thus, the h-method is our preferred approach, for it can locally improve the accuracy of any well-posed numerical scheme.

Before the grid is to be adaptively altered, information about the error distribution must be available, such that discretization is improved only in regions where its resolution is inadequate. This is the role of an error indicator, a proposal for which will now be discussed.

3.1. *Error Estimation*

The convective and diffusive fluxes of momentum, energy, or any other conserved property passing through the cell face are over- or underestimated in some regions as a result of inaccurate assumptions about the spatial variation of the dependent variables. These over- or underestimations can be envisaged as fictitious additional source terms that force the numerical solution to depart from the exact one. These sources are proportional to the truncation error. If the variation of the dependent variable is smooth for a given mesh resolution, it is plausible to assume that the higher powers of mesh spacing h are small multiplicative coefficients in the trunca-

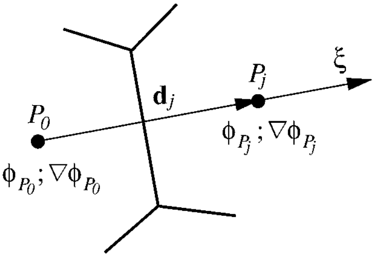
P_0
$\phi_{P_0}; \nabla\phi_{P_0}$
$\mathbf{d}_j$
P_j
$\phi_{P_j}; \nabla\phi_{P_j}$
ξ

FIG. 2. One-dimensional truncation error analysis for variable ϕ.

tion error and that its leading term will be the dominant contributor to the magnitude of the truncation error. Also we can assume that the equations are well posed in the sense that a small change in a term will have only a small effect on the solution, which implies that the solution error will be small if the truncation error is small. While this seems a plausible assumption to make, the authors know of no formal proof for the type of governing equations under consideration.

Some authors (e.g., [5]) have used the difference between upwind and central difference expressions for the convection term to identify the regions in a flow calculation with large numerical diffusion produced by first-order upwind differencing on a given grid. This technique is limited to convection-dominated problems (i.e., the error resulting in modeling diffusion is not considered), and only when convection is modeled by the first-order upwind differencing scheme. Here the idea is extended such that an error estimation is available in case of the use of second-order schemes for both diffusion and convection transport.

The values of variable ϕ and its gradient at two neighboring locations are used in analysis. Instead of looking at the distribution of ϕ in space, the variation only in direction ξ, which connects the neighboring points, is considered (Fig. 2). This simplifies the analysis and reduces the computational cost considerably. Projecting the gradient of ϕ onto the direction $\mathbf{d}_j$, four independent constraints on the variation of ϕ along ξ are available and are given below, enabling calculation of the coefficients of an assumed third-order polynomial variation of ϕ along ξ, i.e.,

$$\phi(\xi) = c_0 + c_1\xi + c_2\xi^2 + c_3\xi^3. \tag{16}$$

The coefficients c_i are to be determined in such a way that (16) satisfies the conditions

$$\begin{aligned} &\xi = 0: \quad \phi = \phi_{P_0}; && \xi = |\mathbf{d}_j|: \quad \phi = \phi_{P_j}; \\ &\xi = 0: (\nabla\phi)^{\xi} = (\nabla\phi)^{\xi}_{P_0}; && \xi = |\mathbf{d}_j|: (\nabla\phi)^{\xi} = (\nabla\phi)^{\xi}_{P_j}; \end{aligned} \tag{17}$$

where $(\nabla\phi)^{\xi}_{P_j}$ is the gradient in direction ξ, i.e.,

$$(\nabla\phi)^{\xi}_{P_j} = (\nabla\phi)_{P_j} \cdot \mathbf{d}_j. \tag{18}$$

The coefficients c_i that satisfy conditions (17) are given by

$$\begin{aligned} c_0 &= \phi_{P_0}, \\ c_1 &= (\nabla\phi)^{\xi}_{P_0}, \\ c_2 &= 3\,\frac{\phi_{P_j} - \phi_{P_0}}{|\mathbf{d}_j|^2} - \frac{(\nabla\phi)^{\xi}_{P_j} + 2(\nabla\phi)^{\xi}_{P_0}}{|\mathbf{d}_j|}, \\ c_3 &= -2\,\frac{\phi_{P_j} - \phi_{P_0}}{|\mathbf{d}_j|^3} + \frac{(\nabla\phi)^{\xi}_{P_j} + (\nabla\phi)^{\xi}_{P_0}}{|\mathbf{d}_j|^2}. \end{aligned} \tag{19}$$

Expression (16) can be used to estimate the values of dependent variable and its gradient in direction $\mathbf{d}_j$ at the cell face j,

$$\begin{aligned} \tilde{\phi}_j &= c_0 + c_1\xi_j + c_2\xi_j^2 + c_3\xi_j^3, \\ \widetilde{\nabla\phi_j}\cdot\frac{\mathbf{d}_j}{|\mathbf{d}_j|} &= c_1 + 2c_2\xi_j + 3c_3\xi_j^2, \end{aligned} \tag{20}$$

where ξ_j is the coordinate that defines the intersection of the distance vector $\mathbf{d}_j$ and the cell face plane. Now it is possible to re-evaluate mass, convection, and diffusion fluxes of a conserved property based on (20) as follows,

$$\tilde{F}_j = \rho_j\tilde{\mathbf{v}}_j\cdot\mathbf{s}_j, \tag{21}$$

$$\tilde{C}_j = \tilde{F}_j\tilde{\phi}_j, \tag{22}$$

$$\tilde{D}_j = -\lambda_{\phi_j}\left(\widetilde{\nabla\phi_j}\cdot\frac{\mathbf{d}_j}{|\mathbf{d}_j|}\right)\frac{\mathbf{d}_j}{|\mathbf{d}_j|}\cdot\mathbf{s}_j, \tag{23}$$

where $\tilde{\mathbf{v}}_j$ is the velocity at the cell face j obtained using (20). In general, these fluxes are different from those obtained during the calculation, since the profile (16) is different from the one used during the discretization described earlier. Summing up the differences between these fluxes over the surfaces bounding the CV leads to an approximation of the source of the truncation error (the so-called tau error [24]) $\tilde{\tau}$ at P_0,

$$\tilde{\tau}_{P_0} = \sum_{j=1}^{n}\left[(\tilde{C}_j - C_j) + (\tilde{D}_j - D_j^N)\right], \tag{24}$$

where the superscript N denotes the normal diffusive flux. The tau error can also be regarded as a source which should be added to the discretized form of the transport equation in order to get the exact solution. Its relationship with the solution error e is given by

$$a_0 e_{P_0} - \sum_{j=1}^{n} a_j e_{P_j} = \tau_{P_0}, \tag{25}$$

where the coefficients a_0 and a_j are the same as those introduced by Eq. (7). Using the approximation $\tilde{\tau}_{P_0}$ of the tau error, the system of Eqs. (25) may be solved to produce an approximation of the solution error e field.

A judgment on which level of the truncation error may be safely neglected, in order to achieve the desired accuracy, will depend on how the tau error is normalized. In the present study, the normalization is

$$\tau^*_{P_0} = \frac{\tilde{\tau}_{P_0}}{a_0 \phi_{\text{ref}}}, \tag{26}$$

where ϕ_{ref} is some reference value of the conserved property ϕ. In this way, according to Eq. (25), the resulting normalized tau error is an approximation to the solution error e, equivalent to performing one Jacobi iteration on this equation. In general ϕ_{ref} might be set to the value of dependent variable at the location where the tau error is calculated, or it can be a typical value in the computational domain or a subdomain. In the present study the normalization is based on a selected typical values of the dependent variables in the considered computational domain.

3.2. *Grid Alteration*

As already noted, the discretization and the data structure presented allow the use of arbitrary polyhedral CVs. The latter in principle can be subdivided in a host of different ways to form smaller polyhedra, so the flexibility carries over to grid refinement strategies. In this study we have chosen to work with meshes which are initially regular hexahedral (with Cartesian as a special case) and then refined by one or more levels of cell subdivision. This particular choice was dictated by the available computer code and time, but as will be seen later it is adequate to demonstrate the capabilities of the methodology.

After solving the discrete system on a given grid, the solution is examined and the cells at which the normalized tau error estimate τ^* for any of conserved properties is above some predetermined value T^* are marked. The boundaries of regions marked for refinement are extended by a safety margin d. The choice of this, together with T^*, determines the part of the domain to be covered by the refined grid components. It is clear that d is related to the characteristic cell size h of the currently finest grid; in the present study it is taken to be $4h$. Each CV marked for grid refinement is subdivided into several smaller CVs in a prescribed manner. A typical hexahedral[2] CV, whose refinement is performed by dividing it into eight (four in 2D) smaller ones, is shown in Fig. 3. Cells next to the boundary are adjusted so that they follow accurately the surface of the original geometry. The grid quality has to be checked after this step since it might happen that after the adjustment very distorted cells are created in regions where the surface geometry exhibits strong curvature and penetrates into several layers of cells of the initial grid (see Fig. 4).

Once a hexahedral cell is locally refined, its nonrefined neighbors effectively have an increased number of faces; i.e., they become "higher-order" polyhedra. This is

[2] It would have been equally possible to use a triangular mesh subdivision: the choice is arbitrary.

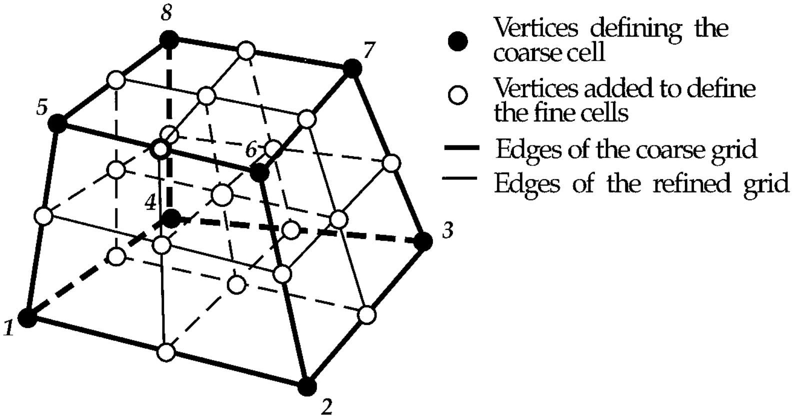
8
7
5
6
4
3
1
2
● Vertices defining the coarse cell
○ Vertices added to define the fine cells
— Edges of the coarse grid
— Edges of the refined grid

FIG. 3. Simple strategy for refinement of a hexahedral control volume.

illustrated in Fig. 5a, which shows an apparently quadrilateral cell *I* adjacent to three refined cells, and Fig. 5b, which shows a topologically equivalent polyhedral cell *J*, here with seven faces. The difference between *I* and *J* is of course that in physical space the former has pairs of coplanar faces. However, in the present methodology fluxes through each of these individual faces are calculated in the same conservative, coupled, implicit manner as fluxes through nonsubdivided faces. The resulting coefficient matrix no longer has the regular banded structure of the base hexahedral mesh, but this also presents no problems to a procedure already geared for unbanded systems.

The above discussion should also make clear that the concept of ‘‘hanging nodes’’ has no place in the present methodology: every cell face is accorded identical treatment, irrespective of the shape of the polyhedron.

The whole procedure of adaptive refinement of the currently finest mesh, and solution of the corresponding discrete system of algebraic equations, is repeated until the error estimates are below a desired value or the available computer memory

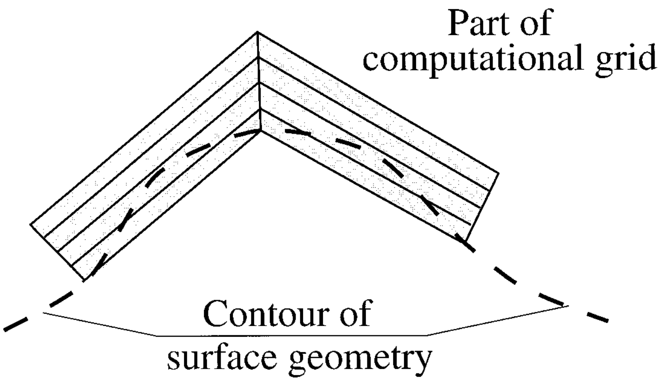

FIG. 4. Large surface curvature and cell aspect ratio.

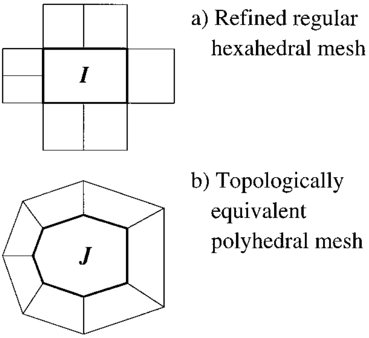
I
a) Refined regular hexahedral mesh
J
b) Topologically equivalent polyhedral mesh

FIG. 5. Two cells of different shape, but identical topology, including neighbor connectivity.

limit is reached. In the early stages of the procedure, when the mesh might be very coarse, the error estimate may not be very reliable. In this case error estimation on finer grids may indicate that previous refinements were not necessary, in which case they can be removed (mesh coarsening). However, the results presented here were produced without mesh coarsening.

4. APPLICATION OF THE METHODOLOGY

In this section the present methodology is applied to four different test cases, one involving laminar flow and the remainder turbulent. The main criteria for their selection were: (a) appreciable degree of flow complexity and (b) availability of benchmark data. Two of the cases involve simple rectilinear geometries for which a Cartesian base mesh is the obvious choice. The other use a body-fitted regular quadrilateral or hexahedral starting mesh.

The first test used to demonstrate the present adaptive refinement methodology is the laminar lid-driven square cavity flow at $\mathrm{Re} = 100$ (Fig. 6). Grid-independent results were initially produced on a uniform 256 × 256 grid using the CD scheme for convection and they agree well with those reported in literature [9, 25]. We use them as the "exact" solution, from which the exact error distribution is derived.

The locally refined grid driven by the present error estimation procedure is shown in Fig. 7. The final grid has 712 CVs and was obtained by four levels of adaptive local refinement, starting from a coarse 8 × 8 grid. All regions where the estimated tau error was larger than 1% with respect to the velocity of the lid U_{lid} were refined. The error was also monitored using Richardson extrapolation and by comparing the solutions on a given grid with the "exact" one. The error estimate is less than the exact error on all grids, but the effectivity index (ratio of exact and estimated errors) does approach unity as the grid becomes finer, as shown in Fig. 8. The error estimation based on Richardson extrapolation shows faster convergence of effectivity index. However, locally refined grids based on both methods have similar distributions of grid points in critical regions (Fig. 7).

The reduction in the number of CVs due to local refinement was 83% compared to the 64 × 64 uniform grid which produced velocity profiles with the same error

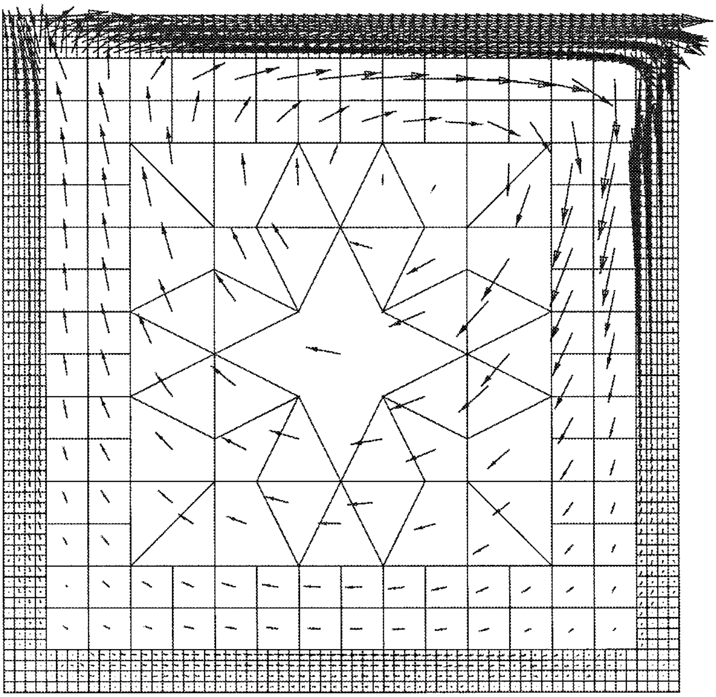

FIG. 6. Cavity-flow velocity field calculated on an unstructured mesh containing cells of different topology.

levels. The velocity profiles obtained on the locally refined grid were in very good agreement with the grid-independent solution, with maximum error of 1.2%.

The second example is the simulation of a turbulent flow through an orifice plate with diameter ratio 0.5 at Re = 18400. The turbulence is modeled using the standard

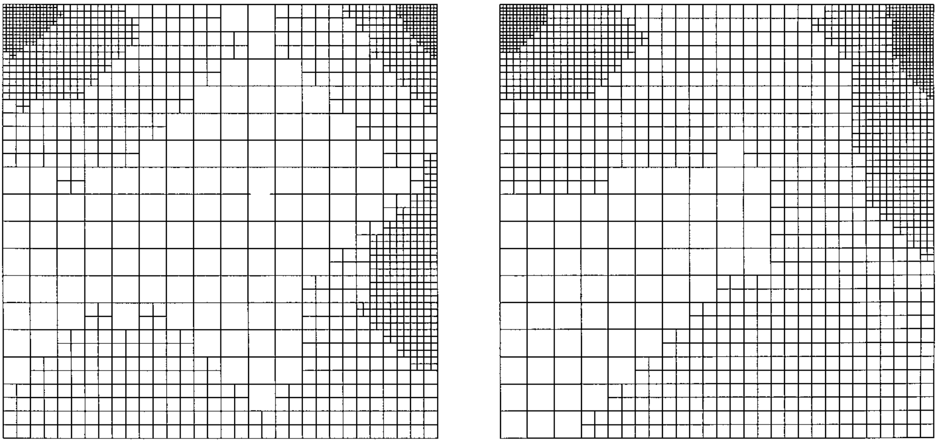

FIG. 7. Locally refined meshes based on Richardson extrapolation (left) and the present method (right).

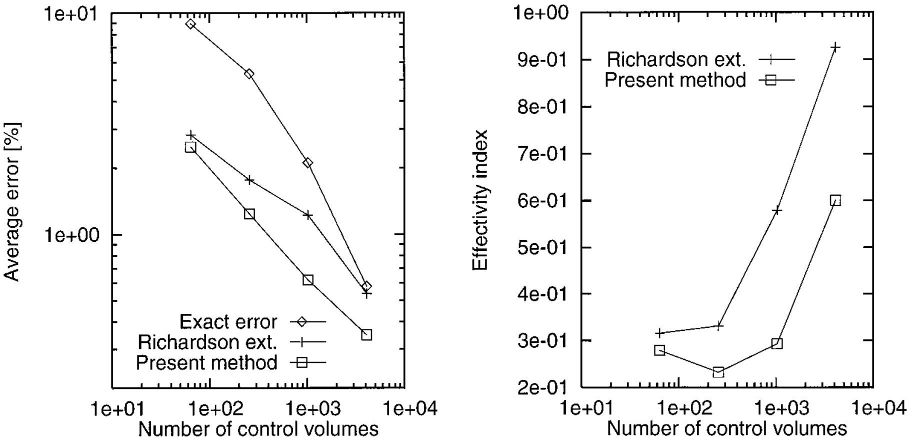

FIG. 8. Averaged errors (left) and corresponding effectivity indices (right).

k-ε model and wall functions. The geometry and boundary conditions were taken from the experiment conducted by Nail [26]. The largest estimated errors for all transported variables occurred, as would be expected, in the region around the orifice, where large gradients of all variables exist. The estimated errors for the axial velocity component were also large in recirculating regions near the wall. Since wall functions are used to model the near-wall effects, the refinement in the near-wall regions was controlled to ensure that values of y^+ fall between the required range of approximately 40 to 80. However, it should be noted that this implies that irreducible discretization errors may remain at near-wall cells. The final locally refined mesh is shown in Fig. 9. The patch of the finest mesh starts just upstream of the orifice plate and extends about six diameters downstream, constantly shrinking toward the axis of symmetry. As can be seen from Fig. 10, this refinement pattern coincides with large spatial variations of the velocity near the axis of symmetry. Figure 10 also shows very good agreement between the predicted and the measured profiles, although in this case this result is also determined by the accuracy of the turbulence model.

For the third case, flow over a surface mounted cube in a wind tunnel is considered. The geometry and boundary conditions were taken from the experiment of Vasilic-Meling [27]. The initial (318 CVs) and a locally refined grid (186452 CVs) are shown in Fig. 11. The available computer memory did not allow sufficient refinement to obtain fully grid-independent results, so it is difficult to judge quantitatively the behavior of the error estimator. The maximum and volume-averaged velocity errors on an intermediate locally refined 61397 CVs mesh were estimated to be 47.26% (near the front and side edges of the obstacle) and 0.65%, respectively (volume averaging was performed by multiplying the local error by the volume of the associated cell, summing these products over the entire computational domain, and dividing by the total volume of the computational domain). The next level of local refinement, performed in regions where the estimated tau error was larger than 1% with respect to the values at the inlet, produced the mesh of Fig. 11 and reduced the maximum and volume-averaged estimated errors to 35.7% and 0.60%,

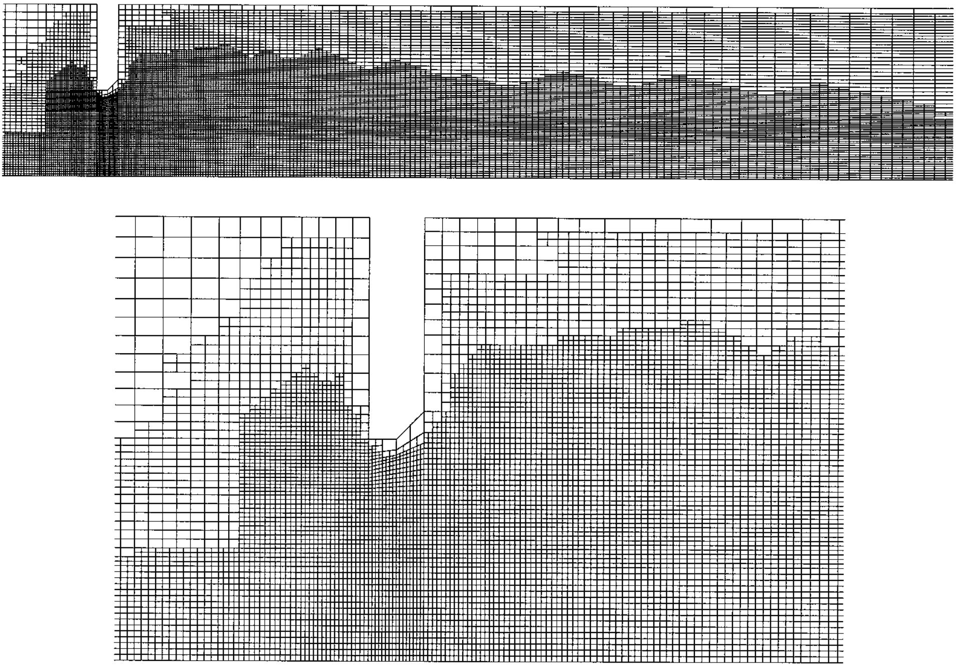

FIG. 9. Details of the error-adapted numerical mesh produced in calculations of the flow through orifice plate.

respectively. Uniform refinement at the finest level would have needed around 1.4 million CVs.

Qualitatively the regions selected for refinement are in accordance with expectations (Fig. 11). The two corners between the walls next to inlet (region A in Fig. 11) are refined since boundary layers develop from the plug inlet velocity profile. The largest spatial variations of all dependent variables are present around the obstacle (region B), demanding higher numerical resolution there. Since the flow is disturbed by the obstacle and redirected toward the side and top walls of the tunnel, large variations are present near the corners and between the bottom surface

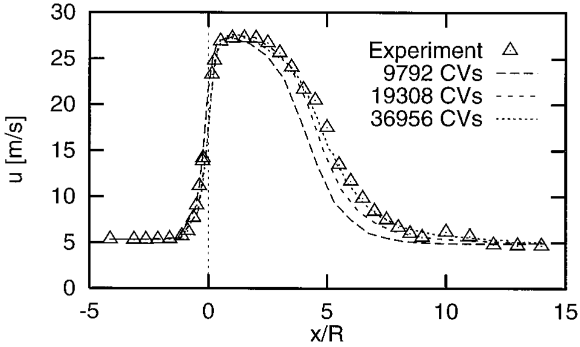

FIG. 10. Mean axial velocity along symmetry axes line as a function of grid resolution.

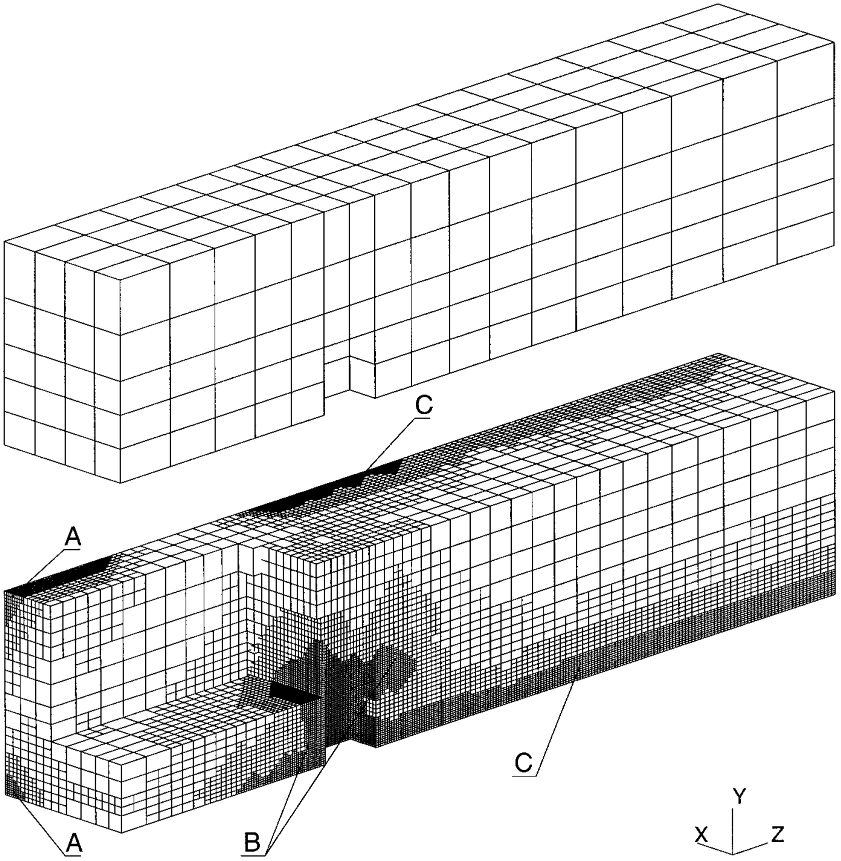

FIG. 11. Initial and final numerical mesh for simulation of the flow around a cubical obstacle.

and the symmetry plane immediately downstream of the obstacle (regions C), as the flow tends toward an undisturbed boundary layer.

Predicted velocity vector and pressure fields in symmetry plane of the channel are shown in Fig. 12. Profiles of predicted local pressure coefficients *Cp* in the same plane and on the surface of the obstacle are compared with the measured ones and presented in Fig. 13. In general, the agreement between measured and predicted profiles is good at the front surface of the obstacle. However, the agreement is poor at the top surface close to the front edge, where the flow separates. The predicted positive pressure gradient in this plane is much larger than the measured one. The trend of this profile is not influenced by the changes of differencing scheme or the numerical resolution, suggesting that differences may be attributed to defects of the turbulence model [28].

The final example is to demonstrate the flexibility of the methodology and the behavior of refinement criterion when applied to complex flows in industrial configurations. The three-dimensional turbulent swirling flow in a water model of a can-type gas turbine combustor, which was the subject of an experimental investigation conducted by Palma [29], is selected for this purpose. The initial mesh (336 CVs) and the final locally refined one (248316 CVs) obtained adaptively refining this one

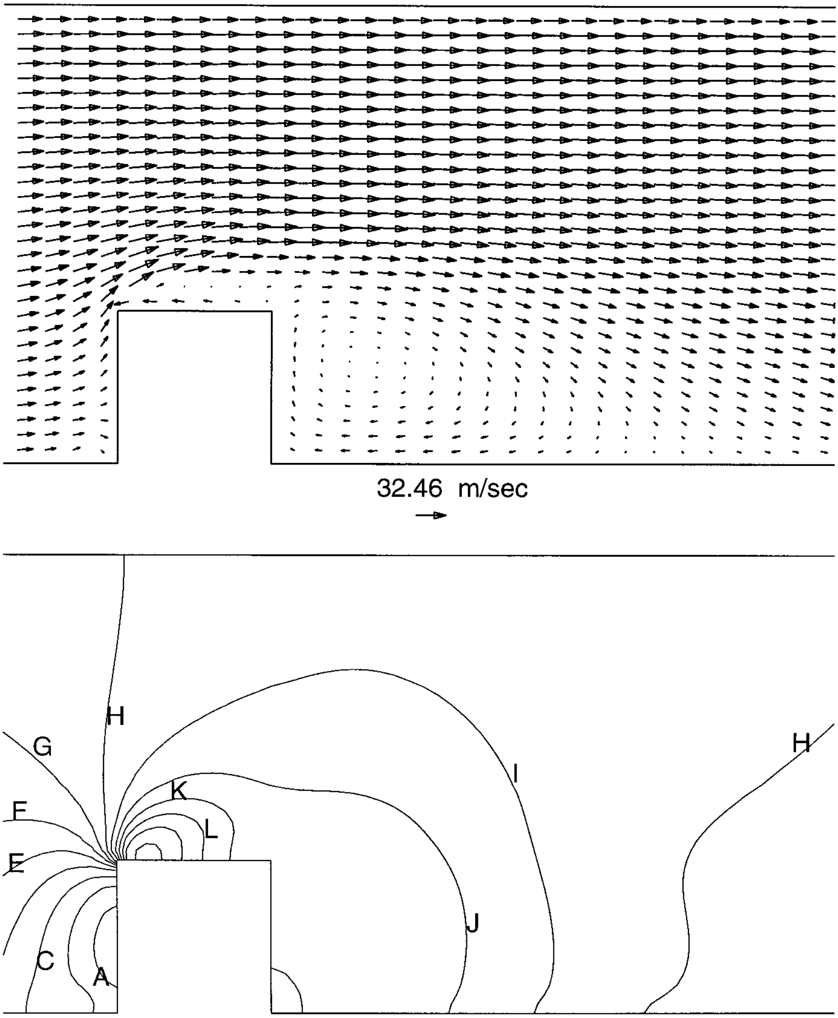

FIG. 12. Velocity and pressure fields in symmetry plane (pressure contours between −457 and 338.1 Pa with step 61.2).

are presented in Fig. 14. The normalization of the refinement measure was based on the values of the dependent variables at the primary holes. Refinement was not performed in the downstream nozzle section of the combustor, since this area was not of interest. Figure 14 gives a qualitative overview of the predicted complex flow field, showing the swirling motion at the inlet, the reverse flow and recirculation zone upstream of the first row of primary jets, the manner in which the axial main flow bypasses the injected jets close to the injection ports, and the intense penetration of the primary jets toward the center line.

A characteristic observed to respond sensitively to both turbulence modeling and numerical resolution is the axial centerline velocity variation, as noted in [30]. Figure 15 depicts four profiles obtained with different grid densities, compared with

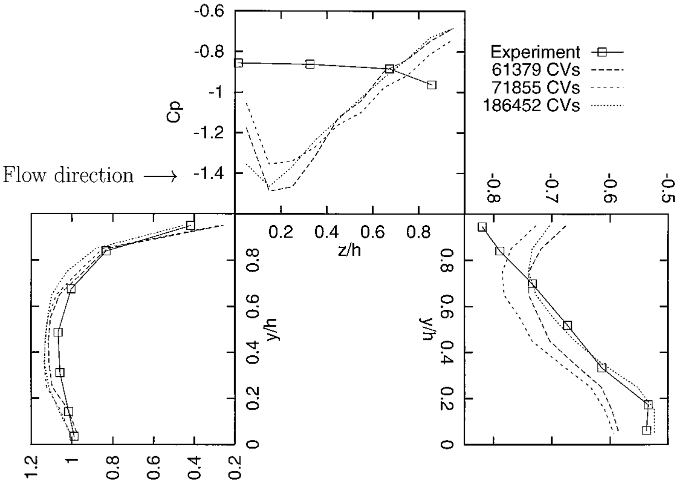

FIG. 13. Distribution of the pressure coefficient around obstacle in the symmetry plane $x = 0$.

the experimental data of Palma [29]. The uniform grids were third and fourth uniform refinements of the initial grid, being composed of 19599 and 153633 CVs, respectively. The locally refined meshes were the result of four and five levels of adaptive refinement. The first locally refined mesh has 88045 CVs, and its finest cells correspond to a mesh resolution of a uniform grid with 153633 CVs. The velocity profiles obtained on these two grids are similar (see Fig. 15). The next level of local grid refinement produced a grid composed of 248316 CVs. The velocity profile obtained on this grid exhibits more strongly an anomalous trough upstream of the primary holes. Lin and Leschziner [30] have linked this to the large axial pressure gradients accompanying a corresponding large axial variation in the swirl velocity in the vicinity of the centerline. They performed calculations for a similar combustor using the standard k-ε eddy-viscosity model and two variants of the Reynolds stress transport (RSTM) model of turbulence. They argued that the k-ε model, being more diffuse, tended to erode the vortex and yielded better agreement with the experimental data than the RSTM models. However, in the present study the increase of the mesh resolution reveals that the k-ε model has poor behavior in this region similar to that of the RSTM.

Since it again was not possible to produce grid-independent results due to computer memory limits (the maximum estimated error on the finest grid was about 65% and the average 9% and their reduction below 1% would need meshes of well over 2×10^6 CVs), it is difficult for this flow to quantitatively judge the results of local grid refinement. However, it can be said that the refinement took place in all regions where one would expect the presence of large discretization errors. More nodes were concentrated in the regions around swirler and primary and dilution jets where the strong shearing flows existed. The relatively uniform cores of the jets were not refined so much, while refinement took place in the mixing layers of the jets where strong velocity variation was present. Refinement also took place in near-wall regions.

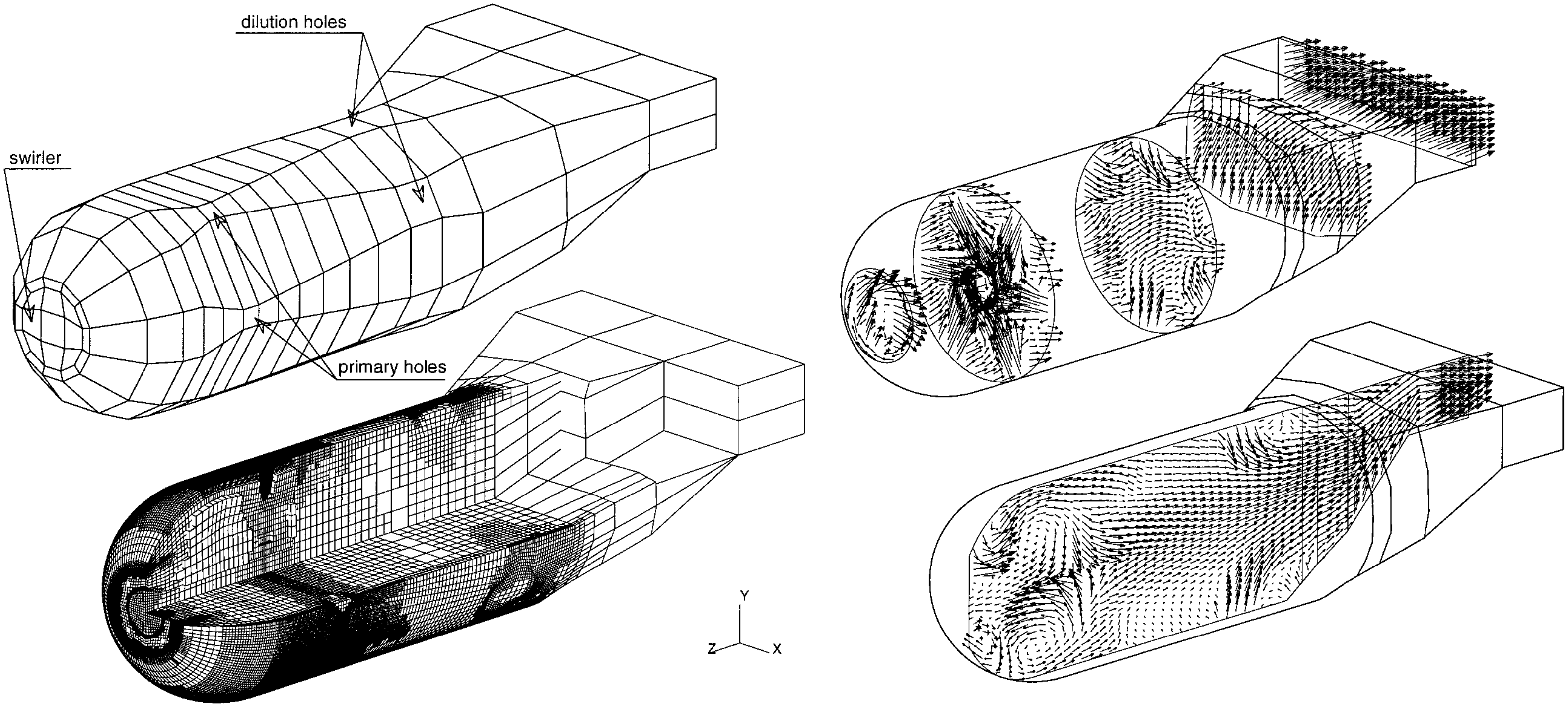

FIG. 14. Initial and final numerical mesh (left) and the overall pattern of the flow inside a turbine combustor (right).

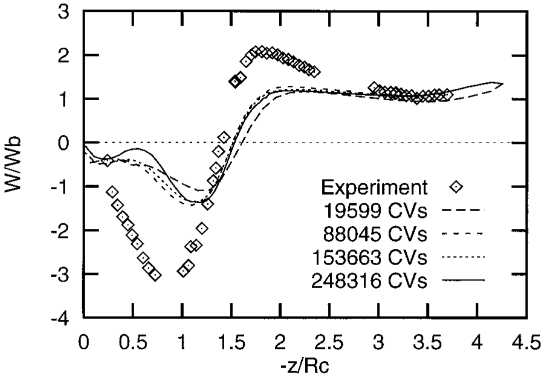

FIG. 15. Axial velocity along the combustor centerline.

The memory and computer times necessary for each of the test cases are summarized in Table I. The table also presents computer times which are obtained by embedding the numerical method described into a full-multigrid algorithm, which will be described in a separate publication.

5. CONCLUSIONS

In this paper we have presented a novel FV discretization practice that does not rely on any particular cell topology. Moreover, it is completely performed in real space avoiding any coordinate transformations, and consequently it is easy to understand, implement, and test. Locally refined grids can be created inexpensively, using simple rules for refinement of existing cells. The polyhedral cells that are created during this process, do not require exceptional treatment.

Local grid refinements controlled by a novel error estimation technique were in accordance with expectations for all cases presented. In the case of the laminar lid-driven cavity flow, the quantitative measures of performance (average estimated error and effectivity index) were satisfactory. However, for the turbulent flow cases full quantitative assessment was not possible due to restriction on grid refinement caused by the use of wall functions and, in the three-dimensional cases, computer resources.

TABLE I

Memory (MB), Number of Iterations, and CPU Times (Minutes) for Each Case on an SGI R3000 Challenge Computer

			Single grid		Multigrid	
Case	No. CVs	Memory	Iter.	CPU	Iter.	CPU
1	916	0.4	552	4.52	32	0.27
2	19308	7.6	960	346.45	152	83.28
3	186452	72.6	631	1798.80	99	396.75
4	248316	95.3	562	2296.46	128	672.08

REFERENCES

1. J. T. Oden and L. Demkowicz, Advances in adaptive improvements: A survey of adaptive finite element methods in computational mechanics, in *State-of-the-Art Surveys in Computational Mechanics*, edited by A. K. Noor and J. T. Oden (ASME, New York, 1989).
2. I. Altas and J. W. Stephenson, *J. Comput. Phys.* **94,** 201 (1991).
3. D. F. Hawken, J. J. Gottlieb, and J. S. Hansen, *J. Comput. Phys.* **95,** 254 (1991).
4. J. J. McGuirk and W. Rodi, *J. Fluid Mech.* **86,** 761 (1978).
5. J. J. McGuirk, A. M. K. P. Taylor, and J. H. Whitelaw, The assessment of numerical diffusion in the upwind-differencing calculations, in *Selected Papers from the Third International Symposium on Turbulent Shear Flow,* edited by L. J. S. Bradbury, F. Durst, B. E. Launder, F. W. Schmidt, and J. H. Whitelaw (Univ. of California Press, Davis, 1983), p. 206.
6. P. Tattersall and J. J. McGuirk, *Comput. Fluids* **23,** 177 (1994).
7. D. C. Haworth, S. H. El Tahry, and M. S. Huebler, *Int. J. Numer. Meth. Fluids* **17,** 75 (1993).
8. M. J. Berger and P. Collela, *J. Comput. Phys.* **82,** 64 (1989).
9. M. C. Thompson and J. H. Ferziger, *J. Comput. Phys.* **82,** 94 (1989).
10. D. M. Smith, Ph.D. thesis, University of London, 1990 (unpublished).
11. S. Caruso, Ph.D. thesis, Thermoscience Division, Department of Mechanical Engineering, Stanford University, 1985.
12. M. J. Berger and J. Oliger, *J. Comput. Phys.* **53,** 484 (1984).
13. J. Y. Tu and L. Fuchs, *Int. J. Numer. Methods Fluids* **15,** 693 (1992).
14. C. Y. Perng and R. L. Street, *Int. J. Numer. Methods Fluids* **13,** 269 (1991).
15. W. L. Chen, F. S. Lien, and M. Z. Leschziner, A local grid refinement scheme within a multiblock structured-grid strategy for general flows, in *6th International Symposium on CFD, Lake Tahoe, USA, September 1995.*
16. R. Vilsmeier and D. Hänel, *Comput. Fluids* **22,** 485 (1993).
17. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).
18. I. Demirdžić, Ž. Lilek, and M. Perić, *Int. J. Numer. Methods Fluids* **16,** 1029 (1993).
19. P. K. Khosla and S. G. Rubin, *Comput. Fluids* **2,** 207 (1974).
20. D. A. Caughey and A. Jameson, Basic advances in the finite volume method for transonic potential flow calculations, in *Numerical and Physical Aspects of Aerodynamic Flows*, edited by T. Cebeci (Springer-Verlag, Berlin, 1982).
21. C. M. Rhie and W. L. Chow, *AIAA J.* **21,** 1525 (1983).
22. J. A. Meijerink and H. A. Van der Vorst, *J. Comput. Phys.* **31,** 134 (1981).
23. H. A. Van der Vorst and P. Sonneveld, ‘‘CGSTAB, A More Smoothly Converging Variant of CGS,’’ Technical Report 90-50, Delft University of Technology, 1990 (unpublished).
24. R. E. Phillips, Ph.D. thesis, Pennsylvania State University, 1984.
25. U. Ghia, K. N. Ghia, and C. T. Shin, *J. Comput. Phys.* **48,** 387 (1982).
26. G. H. Nail, Ph.D. thesis, Texas A&M University, 1991.
27. D. Vasilic-Melling, Ph.D. thesis, University of London, 1976.
28. S.B. Pope and J.H. Whitelaw, *J. Fluid Mech.* **73,** 9 (1976).
29. J. M. L. M. Palma, Ph.D. thesis, University of London, 1988.
30. C. A. Lin and M. A. Leschziner, Computation of three-dimensional injection into swirling combustor-model flow with second-moment closure in *Numerical Methods in Laminar and Turbulent Flow*, edited by C. Taylor, P. Gresho, R. L. Sani, and J. Haüser (Pineridge Press, Swansea, 1989).