INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 375–392 (1997)

ADAPTIVE GRID REFINEMENT USING CELL-LEVEL AND GLOBAL IMBALANCES

S. CHANG AND D. C. HAWORTH

Engine Research Department, GM R&D Center, Warren, MI 48090-9055, U.S.A.

SUMMARY

A methodology for local solution-adaptive mesh refinement in computational fluid dynamics (CFD) using cell-level and global kinetic energy balances is formulated and tested. Results are presented for two two-dimensional steady incompressible laminar benchmark problems: a lid-driven cavity (Reynolds number $Re = 1000$) and a backward-facing step ($Re = 400$). It is demonstrated that local kinetic energy imbalance correlates with local solution accuracy, that normalized global imbalance is an appropriate criterion for halting mesh refinement and that a specified level of accuracy is realized at lower computational effort using local refinement compared with a uniform finer mesh.

KEY WORDS: adaptive mesh refinement; error estimation

1. INTRODUCTION

In diverse applications, three-dimensional time-dependent computational fluid dynamics (CFD) is evolving from research status to engineering tool. In the automotive industry, for example, CFD analysis of in-cylinder flow and combustion processes in reciprocating internal combustion (IC) engines represents an application at the frontier between research and practicable design tool.[1–3] Further progress in the modelling of such complex phenomena demands that numerical accuracy be isolated from physical submodel performance (e.g. turbulence, turbulent combustion and fuel spray models), quantified and controlled. Low-order (first- or second-order) numerical methods generally are selected in these applications for their robustness and computational efficiency. This robustness makes it imperative that there be a means of quantifying the degree to which computed results are faithful to the underlying (modelled) partial differential equations. Also in the event that accuracy is not satisfactory, efficient mechanisms must be available for improving it. Error estimation and reduction should be transparent once a small number of controlling parameters (e.g. maximum acceptable error level and a measure of maximum acceptable CPU time or cost) have been specified. Thus measures of numerical accuracy and strategies for improved accuracy remain a major theme in the CFD literature.[3–5]

Here the focus is on automated solution-adaptive local mesh refinement where mesh density is varied in space and (in time-dependent computations) in time to maintain a specified level of accuracy. Issues include (i) choice of numerical algorithm and data structure, (ii) formulation of local error estimators to establish the spatial distribution of grid points or cells and (iii) formulation of global error measures for determining when the solution is ‘good enough’. Our goal has been to

CCC 0271–2091/97/040375–18

Received 20 April 1995
Revised 28 June 1996

develop an approach that is suited to three-dimensional time-dependent flows in complex geometric configurations including moving boundaries. Here the methodology is evaluated in two two-dimensional steady laminar flow problems: a lid-driven cavity and a backward-facing step.

The basis for the present adaptive mesh refinement scheme is local and global balances of quantities that are not explicitly conserved in the construction of the numerical discretization. The advantages of this approach are several. First, numerical accuracy is assessed from results computed on a single computational mesh: no explicit estimate of the converged solution is required. (Two time levels are needed in transient calculations.) Second, it provides a suitable basis for understanding and controlling numerical error in large-scale engineering computations on coarse meshes. Third, this methodology is compatible with any numerical method (finite difference, finite volume or finite element) using any difference approximations or basis functions. Finally, it is computationally lean: the CPU time required to compute cell-level and global imbalances is less than 1% of the CFD calculation for the pressure based finite volume method employed here.

2. BACKGROUND

We are concerned principally with finite difference and finite volume formulations where mesh density is varied to control local solution accuracy (*h*-refinement). This is in contrast with the use of higher order basis functions (*p*-refinement) that is more prevalent in finite element methods.[6] Moreover, we are interested in mesh refinement via introduction of new grid points ('mesh enrichment') rather than via redistribution of grid points ('mesh movement'). We restrict our attention to body-fitted meshes where solid boundaries coincide with cell faces. This is in contrast with Cartesian mesh approaches for irregular domains that have been presented, for example, by Pember *et al.*[7]

The current scheme bears some similarity to the adaptive mesh refinement algorithm described by Bell *et al.*[8] for block-structured grids. There the emphasis was a hierarchical data structure for embedded regular meshes. Here the focus is the selection and evaluation of error estimators; the current methodology is implemented on unstructured meshes. Richardson extrapolation[9] and solution gradients were used as error estimators in Reference 8 and related work.[7]

2.1. Data structure

Data structures allowing arbitrary connectivity (unstructured meshes) provide a natural framework for incorporating local refinement. Tetrahedral elements are particularly well suited: it is straightforward to introduce or delete grid points, to generate updated connectivity tables and to map dependent variables from one mesh to another.[10] Hexahedral ('brick') elements are more difficult. Here transition regions can be introduced as in Figure 1(a), or algorithms allowing cell splitting can be employed (Figure 1(b)). The former approach maintains a conventional two-cells-per face structure, but transition zone logic becomes cumbersome in three spatial dimensions; cell distortion can also be a problem. Cell splitting requires modifications to conventional data structures, but remains tractable in unstructured codes where pointers (indirect addressing) have already been implemented; cell distortion is controlled by the quality of the parent mesh. A cell-splitting approach for hexahedral elements has been adopted here.

2.2. Local error estimators

Numerical accuracy assessment (and reporting) for engineering applications has generally been limited to heuristic mesh refinement exercises in cases where it has been evaluated at all. Computations are repeated on progressively finer or coarser meshes until a local or global quantity of interest varies little with mesh density. No meaningful quantitative error bounds are established, making it difficult to assess the reliability of the solution.

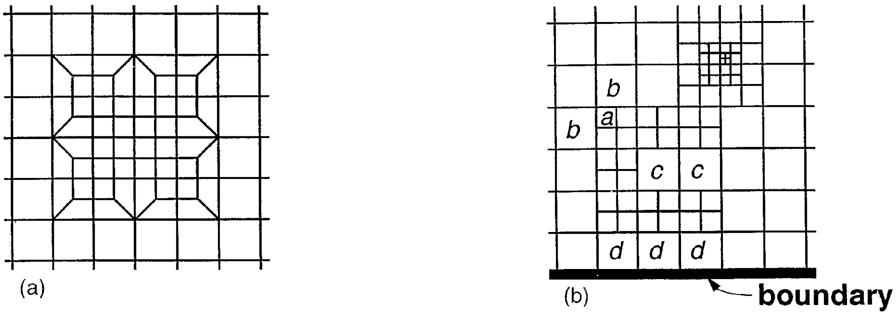

Figure 1. Local mesh refinement in hexahedral element meshes. Two-dimensional examples are shown for clarity. (a) Transition region approach. (b) Cell-splitting approach: cells labelled *a*, *b*, *c*, and *d* illustrate supplementary refinement rules (see text)

Systematic error analysis using Taylor series approximation and Richardson extrapolation provides a sound basis for error estimation in CFD.[4,8,9] In Reference 4, for example, a 'grid convergence index' (GCI) is proposed for the uniform reporting of mesh refinement results. This requires a minimum of two computations on meshes of different densities. Essentially the GCI relates results from a mesh refinement test to the results that one would expect from grid doubling for a second-order method. Alternative approaches to error estimation include residual-based local and global estimators for finite elements.[5,6]

The formal basis for error quantification and control strategies demands that one operate within their asymptotic convergence limits. For example, the leading-order discretization error term in a Taylor series analysis provides a meaningful estimate of convergence rate only in the limit as the grid spacing approaches zero. One cannot know *a priori* whether a given mesh and numerical scheme lie within the asymptotic radius of convergence. In Reference 3, it was demonstrated that mesh refinement and Taylor series extrapolation resulted in misleading estimates of converged solutions even for a relatively simple transient test problem using axisymmetric, orthogonal grids. In practical three-dimensional time-dependent flows, extrapolating to the grid-independent converged solution is more problematic: one can rarely afford to compute on meshes within the radius of convergence.

The difficulty of grid generation and the magnitude of computational resources required for typical industrial flow computations preclude the routine use of grid sensitivity tests and multiple-mesh solutions. For example, a single transient run through intake, compression and combustion on a 300,000-element production port and-cylinder IC engine configuration can require of the order of 100–200 Cray Y-MP hours. Single-run assessments of numerical accuracy and automated accuracy enhancement strategies are needed.

2.3. *Global convergence criteria*

For schemes where an explicit estimate of the converged solution is available (e.g. via extrapolation from solutions obtained on two different meshes), a natural choice for error estimator is the difference between one or more computed dependent variables and their estimated grid-independent converged values. Refinement might continue until the computed dependent variable(s) in every cell lie within a specified tolerance of their converged value(s), and uniform error distribution is achieved. Spatial gradients of pressure or other quantities have been used to control mesh density in applications including flows with shocks or reaction fronts. In this case grid points are added or clustered in regions of steep spatial gradients of the specified dependent variables. Here refinement might continue until a gradient-based length scale is much larger than the local grid spacing.

In Reference 3, global kinetic energy imbalance was reported to be a sensitive indicator of numerical accuracy. The relationship between local solution accuracy and global imbalance was also demonstrated in Reference 3. Kinetic energy is a physically meaningful quantity in many applications of interest. For in-cylinder flows, for example, the kinetic energy of the induction-generated large-scale flow structure provides a 'reservoir' of energy that can be converted into turbulence late during compression. High turbulence translates to more complete mixing and to faster burn. Moreover, shear production via mean velocity gradients is a dominant mechanism for turbulence generation in many internal flow configurations. Length and time scales in physical submodels for turbulent mixing, combustion and fuel sprays are tied to the turbulence scales which in turn are established by the mean flow structure and energy.

Principal issues for this kinetic-energy-imbalance-based error estimator are its generality, and the relationship between kinetic energy imbalance and other physical quantities of interest. These are discussed in Section 7.

3. BASE FLOW SOLVER

The flow solver accepts an unstructured mesh of hexahedral 'brick' elements. Dependent variables are located at cell centres, and each grid point or node is shared by an arbitrary number of elements. The principal equations solved are the Favre-averaged equations for momentum, pressure (continuity), internal energy, mass fraction, turbulence kinetic energy and the viscous dissipation rate of turbulence kinetic energy (the last two for turbulent flows). A pressure-based algorithm patterned after SIMPLE[11] is adopted, where pressure is computed by enforcing continuity and density is specified via an equation of state. Centred second-order differences are used for diffusion terms, while a blend of upwind and central differencing is implemented for the convective terms in the momentum equations. For steady flow problems there is provision for time marching to the steady solution or for iterating using under relaxation.

For the steady, constant-property, laminar flows that are the subject of the present investigation, the continuity and momentum equations reduce to

$$\frac{\partial U_i}{\partial x_i} = 0, \qquad \frac{\partial \rho U_i U_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j}, \tag{1}$$

with

$$\tau_{ji} = \mu\left(\frac{\partial U_j}{\partial x_i} + \frac{\partial U_i}{\partial x_j}\right). \tag{2}$$

Here U_i refers to the ith Cartesian component of velocity, p is the pressure, τ_{ji} is the viscous stress tensor and ρ and μ are the constant fluid density and viscosity respectively. Summation is implied over repeated italic indices.

By design of the finite volume method, mass and linear momentum are conserved for each cell in the discrete formulation. Angular momentum components, mean kinetic energy and other quadratic and higher-order quantities are not conserved. For any *convergent* discretization,[12] however, these quantities should also be conserved in the limit as the grid spacing approaches zero, i.e. in the limit of a grid-independent converged solution. Thus these non-conserved quantities are potential candidates for assessing numerical inaccuracy.

4. REFINEMENT STRATEGY

4.1. Kinetic energy budget

Balance equations for linear momentum, angular momentum, mean flow kinetic energy and turbulence kinetic energy can be found in References 3 and 13. There the equations have been derived in a form suitable for time-dependent compressible turbulent flows in regions with moving boundaries.

For clarity we restrict attention to steady, laminar, incompressible flows; this is not a fundamental limitation of the methodology. In laminar flow the turbulence kinetic energy is equal to zero and we drop the distinction between mean and instantaneous quantities. The total kinetic energy in an arbitrary volume V is $\hat{K} \equiv \int_{\mathrm{V}} \frac{1}{2}\rho U_j U_j \mathrm{d}\mathscr{V}$. The kinetic energy budget for an arbitrary volume V with bounding surface S is derived by taking the inner product of the momentum equation in (1) with the velocity vector, integrating over V and manipulating the result to obtain[3,13]

$$\dot{\hat{K}}_{\mathrm{num}} = \int_S \tfrac{1}{2}\rho U_j U_j U_l \mathrm{d}\mathscr{S}_l + \int_S p U_j \mathrm{d}\mathscr{S}_j - \int_S \tau_{ji} U_j d\mathscr{S}_i + \int_V \tau_{ji}(\partial U_j/\partial x_i)\mathrm{d}\mathscr{V}. \tag{3}$$

Here $\mathrm{d}\mathscr{S}_j$ is the jth component of the outward-pointing area element vector. The term $\dot{\hat{K}}_{\mathrm{num}}$ is the numerical imbalance in kinetic energy for the volume V resulting from replacing partial differential equations (1) and (2) with difference approximations.

Equation (3) expresses the budget of kinetic energy over a domain V that can vary from a single computational cell to the entire computational domain. From left to right the terms on the right-hand side of equation (3) represent the rate at which kinetic energy is advected out of V through S (*FLUX*), the rate at which pressure forces on S extract kinetic energy from V (*PRES*), the rate at which viscous stresses over S extract kinetic energy from V (*SHEAR*) and the rate at which viscous stresses convert kinetic energy to sensible energy (heat) over the interior of the volume V (viscous dissipation, *DISS*). This budget can be written symbolically as

$$\dot{\hat{K}}_{\mathrm{num}} = \mathrm{FLUX} + \mathrm{PRES} + \mathrm{SHEAR} + \mathrm{DISS}. \tag{4}$$

The cell-level kinetic energy imbalance will be denoted by $\dot{\hat{K}}_{\mathrm{num,\ c}}$ and the global imbalance over the entire computational domain by $\hat{K}_{\mathrm{num,\ g}}$. Additional terms arise in unsteady compressible turbulent flows.[3,13]

4.2. Implementation details

In the event that a cell's error estimator (here $\dot{\hat{K}}_{\mathrm{num,\ c}}$) exceeds a threshold value, the 'parent' cell is split into several 'child' cells. Isotropic cell splitting proceeds by introducing one new node at the centroid of the cell volume and one new node at the centre of each face and edge of the parent cell (Figure 1(b)). Thus for every refined parent cell there are n_{c} child cells,

$$n_{\mathrm{c}} = 2^D, \tag{5}$$

where D is the number of spatial dimensions ($D = 1$, 2, or 3).

Parameters are introduced to establish the average rate at which computational cells proliferate during refinement and to establish global convergence criteria for halting refinement. A 'growth rate' R_{c} is introduced as the ratio of the number of cells after one refinement event, $N_{\mathrm{c,\ after}}$ to the number of cells before the event, $N_{\mathrm{c,\ before}}$. This ratio can be expressed as a function of the number of child cells, n_c (equation (5)), created from each parent and of the fraction of cells split at each refinement level, f. Here the cells to be split are the $f \cdot N_{\mathrm{c,\ before}}$ cells having the highest cell-level kinetic energy imbalance magnitude $|\hat{\dot{K}}_{\mathrm{num,\ c}}|$. If n_c is the same for all parent cells (as in the present scheme), then

$$R_c = \frac{N_{\mathrm{c,\ after}}}{N_{\mathrm{c,\ before}}} = 1 + (n_c - 1) f. \tag{6}$$

In practice, f is selected to control the growth rate R_c according to equation (6) with n_c from equation (5). Because the growth rate is limited, a poor initial mesh might result in persistent errors in transient calculations.

There is no inherent restriction on the number of cells that share a common face. Here this number has been limited to 2^{D-1} to make the data structure more efficient. For instance, in the two-dimensional example of Figure 1(b), if cell *a* is tagged for splitting, then all cells *b* will also be split regardless of the value of cell *b*'s error estimator. In addition, any cell that lies between two tagged cells (cells *c* in Figure 1(*b*)) or any cell that lies between a tagged cell and a boundary (cells *d* in Figure 1(b)) is also tagged for refinement. These supplementary rules are intended to maintain smooth transitions between refined and unrefined regions for cells whose error estimators are close to the threshold value.

A global convergence criterion or 'stopping criterion' is established by applying the kinetic energy balance of equation (4) to the entire computational domain. This is equivalent to summing the cell-level balances over all computational cells. The result is a measure of the global numerical imbalance in kinetic energy, $\hat{\dot{K}}_{\mathrm{num,\ g}}$. A physically meaningful convergence criterion is that the numerical loss in kinetic energy should be a small fraction q of a reference kinetic energy rate $\hat{\dot{K}}_{\mathrm{ref}}$, where the latter can be the rate of kinetic energy supplied to the system or the physical dissipation rate *DISS*:

$$\overline{IMBAL_g} \equiv |\hat{\dot{K}}_{\mathrm{num,\ g}} / \hat{\dot{K}}_{\mathrm{ref}}| < q. \tag{7}$$

Parameter values adopted here are $f = \frac{1}{3}$ and $q = 0{\cdot}05$. The former corresponds (nominally) to a doubling of the number of cells with each refinement in two spatial dimensions ($D = 2$). The latter means that not more than 5% of the reference kinetic energy should be lost numerically.

In summary, solution-adaptive local mesh refinement proceeds in five steps. 1) Set global control parameters: the growth rate R_c or f (equation (6)) and the maximum acceptable error q (equation (7)). 2) Compute a numerical solution on the current mesh. 3) From the current (converged or partially converged) solution, compute the cell-level error estimator $\hat{\dot{K}}_{\mathrm{num,\ c}}$ (equation (4)) and normalized global imbalance $\overline{IMBAL_g}$ (equation (7)). 4) Refine the mesh based on the error estimator and supplementary cell connectivity rules. 5) Repeat steps 2)–4) until global convergence is satisfied (equation (7)) or auxiliary limits (total number of cells, CPU time) are reached.

5. TEST CASES

5.1. Lid-driven cavity

This configuration (Figure 2(a)) has been the subject of exhaustive numerical studies.[14,15] Dimensional parameters for the two-dimensional problem are cavity dimensions $h \times h$, wall speed U_w and fluid properties ρ and μ. Here computations have been done for a Reynolds number $Re \equiv \rho U_w h / \mu = 1000$. The solution proceeds iteratively to steady state using underrelaxation starting from a quiescent field. Computations have been performed on a three-dimensional mesh with a single cell of span w in the z-direction. Residuals were normalized with $\rho U_w w h$ for the continuity equation and $\rho U_w^2 w h$ for the momentum equations. The solution is considered to be converged in the iterative

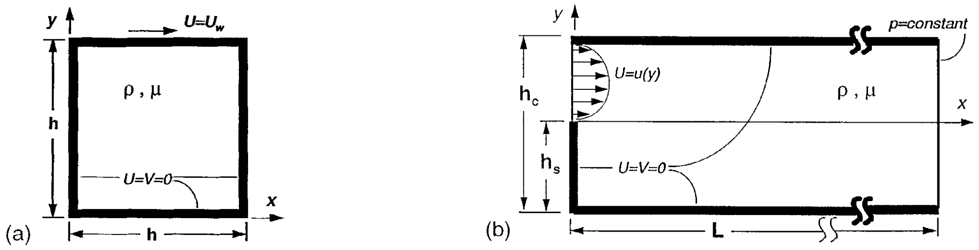

Figure 2. Computational configurations: (a) lid-driven cavity; (b) backward-facing step

steady state sense when normalized residuals are smaller than 10^{-4}. Alternatively, one might monitor convergence of *IMBAL* directly; here we have opted to retain the standard practice for iterative pressure-based solution procedures.

The stopping criterion is taken to be the global rate of kinetic energy loss normalized by the rate of kinetic energy input at the driven wall, *SHEAR*:

$$\dot{\hat{K}}_{\text{ref}} = |SHEAR| = \left| \int_{\text{driven wall}} \tau_{ji} U_j \, \mathrm{d}\mathscr{S}_i \right|. \tag{8}$$

Because there is no flux across solid boundaries ($FLUX = 0$) and no component of wall velocity normal to the walls ($PRES = 0$), the global kinetic energy imbalance over the entire computational domain reduces to $\hat{K}_{\text{num, g}} = SHEAR_{\text{g}} + DISS_{\text{g}}$ (equation (4)). Thus normalizing by the rate of energy input at the driven wall is essentially equivalent to normalizing by the rate of viscous dissipation: $SHEAR = -DISS$ in the limit of a grid-independent converged solution.

Benchmark velocity profiles generated by Ghia *et al.*[14] (129 × 129 mesh, central differencing) are found to be indistinguishable from the present results computed on a uniform 160 × 160 mesh using central differencing. For convenience the latter computations are referred to as the benchmark in Section 6.

5.2. *Backward-facing step*

This two-dimensional configuration (Figure 2(b)) has likewise been the subject of considerable attention.[15–18] Dimensional parameters are channel height h_{c}, step height $h_{\text{s}} = h_{\text{c}}/2$, length of computational domain, L, inlet velocity profile $u(y)$ and fluid properties ρ and μ. The velocity scale is taken to be the area-averaged inlet velocity $\bar{U}_{\text{b}} = \int_{\text{inlet}} u(y)\mathrm{d}y/(h_{\text{c}} - h_{\text{s}})$. Here the Reynolds number is $Re \equiv \rho \bar{U}_{\text{b}} h_{\text{c}}/\mu = 400$.

A parabolic velocity profile is specified at $x = 0$ (Figure 2(b)) and a uniform pressure across the outlet. The stability of this two-dimensional laminar flow has been established recently for Reynolds numbers up to 800.[18] We were able to obtain steady state solutions at $Re = 800$ only via time-consuming transient marching to the steady solution. Thus we have selected a lower Reynolds number of 400 to expedite our numerical tests. For the backward-facing step the mesh expands in the streamwise (x) direction even for regular mesh cases. Residuals were normalized with $\rho \bar{U}_{\text{b}} w h_{\text{c}}$ for the continuity equation and $\rho \bar{U}_{\text{b}}^2 w h_{\text{c}}$ for the momentum equations, where w is the channel width in the z-direction.

In this case the rate of numerical kinetic energy loss has been normalized by the net rate of kinetic energy influx (inlet flux minus outlet flux). For comparison we also report results normalized by the viscous dissipation rate:

$$\dot{\hat{K}}_{\mathrm{ref},\,1} = \left| \int_S \frac{1}{2} \rho U_j U_j U_l \, \mathrm{d}\mathscr{S}_l \right|, \qquad \dot{\hat{K}}_{\mathrm{ref},\,2} = \left| \int_V \tau_{ji} (\partial U_j / \partial x_i) \, \mathrm{d}\mathscr{V} \right|. \tag{9}$$

These yield comparable numerical values as well as trends, since *SHEAR* and *PRES* are small for this configuration. (The only non-zero contributions to these two terms are over the inflow and outflow boundaries.) We adopt as the benchmark solution our most accurate numerical solution: 800 × 80 cells with central differencing.

6. RESULTS

The run matrix is summarized in Table I and results are given in Tables II and III. There 'LDC' designates lid-driven cavity cases while 'BFS' denotes backward-facing step runs; an 'R' in the case name denotes local refinement. Two sets of computations are reported for each configuration, one

Table I. Run summary. 'LDC' designates lid driven cavity runs, and 'BFS' denotes backward-facing-step runs. An 'R' in the case name indicates local refinement and the final 'U' or 'C' distinguishes upwind from central differencing. N_c is the total number of computational cells. Refinement is controlled by the parameter f defined by equation (6); here $f = \frac{1}{3}$

Case	N_c	Convective differencing	Description
LDC1U	400	Upwind	20 × 20 uniform
LDC2U	1600	Upwind	40 × 40 uniform
LDC3U	6400	Upwind	80 × 80 uniform
LDC4U	25600	Upwind	160 × 160 uniform
LDCR1U	838	Upwind	Refinement 1, from LDC1U
LDCR2U	1756	Upwind	Refinement 2, from LDCR1U
LDCR3U	3778	Upwind	Refinement 3, from LDCR2U
LDC1C	400	Central	20 × 20 uniform
LDC2C	1600	Central	40 × 40 uniform
LDC3C	6400	Central	80 × 80 uniform
LDC4C	25600	Central	160 × 160 uniform (benchmark)
LDCR1C	832	Central	Refinement 1, from LDC1C
LDCR2C	1912	Central	Refinement 2, from LDCR1C
LDCR3C	4339	Central	Refinement 3, from LDCR2C
BFS1U	1000	Upwind	100 × 10 regular
BFS2U	4000	Upwind	200 × 20 regular
BFS3U	16000	Upwind	400 × 40 regular
BFS4U	64000	Upwind	800 × 80 regular
BFSR1U	1981	Upwind	Refinement 1, from BFS1U
BFSR2U	4012	Upwind	Refinement 2, from BFSR1U
BFS1C	1000	Central	100 × 10 regular
BFS2C	4000	Central	200 × 20 regular
BFS3C	16000	Central	400 × 40 regular
BFS4C	64000	Central	800 × 80 regular (benchmark)
BFSR1C	2077	Central	Refinement 1, from BFS1C
BFSR2C	4693	Central	Refinement 2, from BFSRlC

using upwind differencing and the other central differencing (with deferred correction) for convective terms in the momentum equations. Underrelaxation factors for pressure and momentum are $\alpha_p = 0{\cdot}3$ and $\alpha_u = 0{\cdot}7$ respectively. No attempt has been made to optimize α_p and α_u. The number of cells does not exactly double with each refinement (e.g. BFSR2C versus BFSR1C) because of the supplementary refinement rules imposed to ensure mesh quality. Here 'doubling' the mesh corresponds to a refinement ratio of $\sqrt{2}$ ($\approx 41\%$) in each direction.

For evaluation purposes the CPU time and iteration counts reported in Tables II and III reflect the computational effort for that mesh starting from quiescent initial conditions. In applications we begin with a coarse mesh and refine periodically, computing the error estimator based on a partially converged solution and taking the latest available solution as the initial condition for each successive refinement.

6.1. *Lid-driven cavity*

Results for four uniform meshes and for six meshes with local refinement are presented in Plate 1, Figures 3 and 4 and Table II. The benchmark ('exact') velocity field (LDC4C) is denoted by $\mathbf{U}_{\rm e} = (U_{{\rm e},\, x}, U_{{\rm e},\, y})$. Local and global deviations between computed and benchmark velocities are defined as

$$\begin{aligned}
\Delta U_{\rm e,c}^{(i)} &\equiv \{[(U_x^{(i)} - U_{{\rm e},x}^{(i)})^2 + (U_y^{(i)} - U_{{\rm e},y}^{(i)})^2]/(U_{{\rm e},x}^{(i)2} + U_{{\rm e},y}^{(i)2}2)\}^{1/2} \quad \text{(for cell } i\text{)},\\
\Delta U_{\rm e,g} &\equiv \frac{1}{N_{\rm c}} \sum_{i=1}^{N{\rm c}} \Delta U_{\rm e,c}^{(i)} \quad \text{(global)}.
\end{aligned} \tag{10}$$

Plate 1 illustrates the significance of the choice of error estimator. The cell-level kinetic energy imbalance (Plate 1(a)) is the error estimator adopted here. This error distribution is strongly peaked at large magnitudes of the imbalance; that is, a small number of cells have very large imbalance magnitudes. Imbalance magnitude ranges from 3×10^{-13} to 3×10^{-8} in Plate 1(a) with the high-error cells concentrated in the upper corners near the intersection of fixed and driven walls. Normalizing the cell-level kinetic energy imbalance by the cell level kinetic energy $\hat{K}_{\rm c}$ (Plate 1(b)) emphasizes regions where the flow has little energy locally, including the recirculation zones in the two lower corners of the cavity and the centre of the main vortex. A velocity gradient-based error estimator (Plate 1(c)) highlights the upper and right-hand walls, consistent with the kinetic energy imbalance. In addition, it flags the separation streamline for the recirculation zones in the two lower corners. Normalized departure from the benchmark solution (Plate 1(d)) marks essentially the same low-energy regions as the normalized kinetic energy imbalance of Plate 1(b).

Mesh distributions for three levels of refinement starting from LDC1U are shown in Figures 3(a)–3(c). With each successive refinement the error becomes more uniform. By the third level (LDCR3U), only a few isolated patches of high error remain along the driven wall near the corners. With central differencing, a different distribution of cells is tagged for refinement at each level, resulting in the third-level mesh pattern of Figure 3(d). For both differencing schemes, refinement is concentrated close to the driven wall and down the right-hand ($x = h$) wall, with 'fingers' extending from the right-hand wall into the body of the flow. Central differencing yields more refinement adjacent to the lower and left-hand fixed walls.

Table II reports global error estimates and computational effort for lid driven cavity cases. The global rate of numerical loss of kinetic energy, $\hat{K}_{{\rm num},\ {\rm g}}$ (column (b)), generally decreases with increasing spatial resolution and with higher-order differencing. A sensible normalization is important: the rate at which kinetic energy is supplied by the driven wall, $\hat{K}_{\rm ref} = |SHEAR|$, increases as the grid is refined in the vicinity of the wall. Thus there is a higher rate of supply of kinetic energy to the system, a higher rate of physical dissipation of kinetic energy and potentially a higher rate of numerical kinetic energy loss as the mesh is refined. It is the fraction of available energy that is lost numerically that is relevant for assessing accuracy. This fraction decreases monotonically even in cases where the energy loss itself increases (e.g. LDC1U versus LDCR1U). None of the upwind cases achieves a 5% or less energy loss, while two of the central differencing cases (LDC4C and LDCR3C) satisfy this criterion.

Table II also reveals that the normalized global kinetic energy imbalance ranking (equation (7), column (d)) is nearly the same as the ranking based on global error with respect to the benchmark velocity field (equation (10), column (e)). This consistency is further demonstrated in Figure 4. There, monotonic convergence to the benchmark velocity profiles is evident with refinement on uniform meshes. Moreover, the relative accuracy implied by the deviation from the benchmark solution in Figure 4 is consistent with the ranking implied by the percentage of energy loss in Table II. For

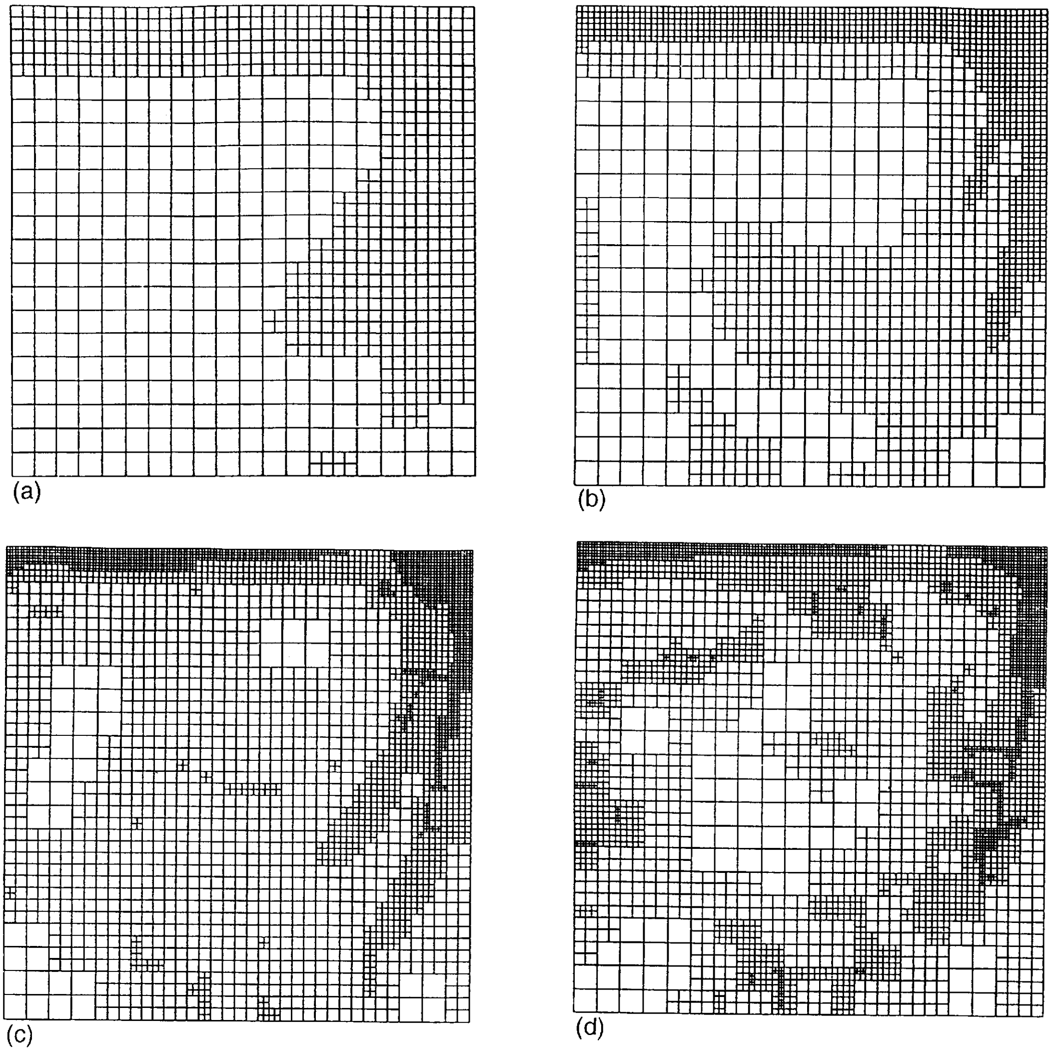

Figure 3. Mesh patterns starting from a uniform 20 × 20 mesh, lid-driven cavity: (a) LDCR1U; (b) LDCR2U; (c) LDCR3U; (d) LDCR3C

example, LDCR3C x- and y-velocity profiles are somewhat closer to the benchmark solution than LDC3C profiles, consistent with the global error measures of Table II.

Compared with upwind, the overhead for central differencing is a factor of two to three in CPU time for a given mesh size N_c (Table II). However, to reduce the error to the same level by increasing N_c with upwind differencing is much more costly. For example, a 10% energy loss level ($\overline{IMBAL}_g \leqslant 0{\cdot}10$) is realized with central differencing for a normalized CPU time of about 100 on a uniform mesh (LDC3C) compared with a CPU time close to 10 using local refinement (LDCR2C). With upwind differencing, the 10% error level requires more than 1000 CPU units (LDC4U) on a uniform mesh.

A final observation is that local refinement is computationally efficient. For example, cases LDC2C and LDCR2C have approximately the same number of computational cells (≈ 1600) and comparable CPU times and iteration counts. However, the normalized energy loss (column (d), Table II) for the former is nearly double that of the latter. Equivalently, a given level of accuracy can be realized with lower computational effort using local refinement. Here cases LDC3C and LDCR2C show similar normalized energy losses (about 10%), but the number of computational cells for the former is about four times that of the latter and the CPU time advantage with local refinement is a factor of nine. Efficiency benefits of local refinement are even greater when refining based on partially converged solutions. Examples are given in column (g) of Table II: for LDCR2C the CPU time is reduced by about 20% (10·76 versus 12·8).

Table II. Lid-driven cavity results. Iteration counts and CPU times are normalized by values for the coarsest uniform mesh with upwind differencing. Numbers in parentheses rank the solutions from most accurate (1) to least accurate (7), for upwind and for central differencing independently. CPU time ratios in square brackets (column (g)) were obtained using fully automated adaptive meshing starting from a coarse mesh, with error estimators computed using partially converged solutions

Case	$\lvert 10^8 \dot{\hat{K}}_{\mathrm{num,g}} \rvert$ (equation (4)) ($\mathrm{kg\ m^2\ s^{-3}}$)	$\lvert 10^8 \dot{\hat{K}}_{\mathrm{ref}} \rvert$ ($\lvert SHEAR \rvert$, equation (8)) ($\mathrm{kg\ m^2\ s^{-3}}$)	$\overline{IMBAL}_{\mathrm{g}} \times 100$ (equation (7)) (%)	$\Delta U_{\mathrm{e,g}}$ (equation (10))	Iteration count ratio	CPU time ratio
(a)	(b)	(c)	(d)	(e)	(f)	(g)
LDC1U	3·90	9·43	41·3 (7)	0·71 (7)	1·00	1·00
LDC2U	3·89	12·0	32·3 (5)	0·42 (5)	2·92	4·85
LDC3U	2·90	13·6	21·3 (3)	0·27 (3)	10·5	59·2
LDC4U	1·83	14·7	12·4 (1)	0·19 (1)	40·1	1208
LDCR1U	3·93	12·1	32·5 (6)	0·49 (6)	1·55	2·00
LDCR2U	3·08	13·7	23·4 (4)	0·33 (4)	2·49	4·65
LDCR3U	2·18	14·9	14·7 (2)	0·23 (2)	4·16	16·03 [10·12]
LDC1C	2·82	8·85	31·9 (7)	0·47 (7)	2·85	2·85
LDC2C	2·21	11·2	19·7 (5)	0·14 (5)	6·78	11·4
LDC3C	1·23	12·9	9·60 (3)	0·036 (3)	18·8	106·9
LDC4C	0·51	14·2	3·63 (1)	0·00 (1)	55·8	1674
LDCR1C	2·27	11·2	20·2 (6)	0·22 (6)	3·81	4·91
LDCR2C	1·30	12·9	10·1 (4)	0·080 (4)	6·12	12·8 [10·76]
LDCR3C	0·60	14·1	4·25 (2)	0·031 (2)	10·6	45·7

6.2. Backward-facing step

Refinement patterns for upwind and central differencing are displayed in Figure 5. Refinement is concentrated adjacent to the inflow boundary, along the entry region of the upper wall and along the separation streamline in the high-shear region. Again, different error estimators highlight different flow features (not shown).

The location of the reattachment point generally moves downstream with improving numerical accuracy. The benchmark solution yields a reattachment length of 4·30 h_c, while Reference 15 reported a value of 4·35h_c.

Global error estimates and computational effort are summarized in Table III. Here, normalization of the numerical kinetic energy imbalance is less important than for the lid-driven cavity configuration: the rate of kinetic energy supply varies little with mesh density or differencing scheme.

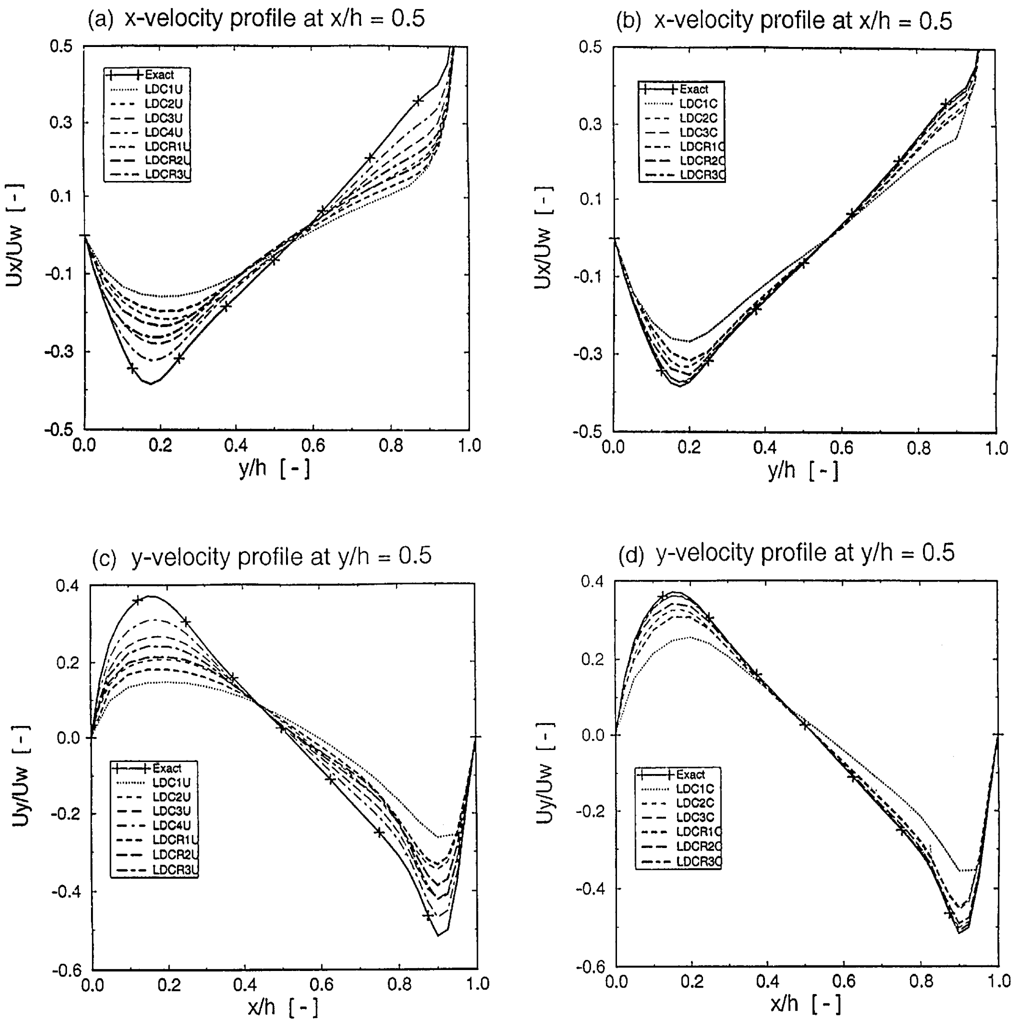

Figure 4. Normalized velocity profiles for lid-driven cavity: (a) *x*-velocity, seven upwind cases plus benchmark ('exact'); (b) *x*-velocity, seven central cases; (c) *y*-velocity, seven upwind cases plus benchmark ('exact'); (d) *y*-velocity, seven central cases

It still is appropriate to express the rate of numerical energy loss as a percentage of a reference rate to establish a physically meaningful global convergence criterion. Two possibilities are reported in Table III: normalization by the net influx of kinetic energy (column (e)) and normalization by the physical rate of kinetic energy dissipation (viscous dissipation, column (f)). These two values are very close to one another. In the limit of a grid-independent solution (no numerical losses of kinetic energy) and with $SHEAR \approx 0$ and $PRES \approx 0$ the global energy balance (equation (4)) reduces to $FLUX + DISS = 0$.

The normalized global kinetic energy loss (equation (7)) generally ranks the runs in the same order as the global solution deviation (equation (10)). Departures of computed profiles from the benchmark profiles likewise yield a consistent ranking (Figures 6 and 7: the $x/h_c = 3$ station cuts through the recirculation bubble, while $x/h_c = 7$ lies downstream of the reattachment point). An exception is BFS1U versus BFSR1U. There $U_{e,g}$ increases slightly from BFS1U to BFSR1U, while the normalized energy loss and convergence of velocity profiles suggest that the latter case is more accurate. This discrepancy may result from the solution behaviour far downstream in the computational domain.

For quiescent initial conditions the efficiency benefit of local refinement is less clear than for the lid-driven cavity. The 2000-cell non-uniform meshes (BFSR1U and BFSR1C) yield global errors that are smaller than those of the respective 1000-cell uniform meshes (BFS1U and BFS1C) and are closer to the 4000-cell uniform mesh values (BFS2U and BFS2C). Computational times for the 2000-cell non-uniform meshes are less than half those of the corresponding 4000-cell uniform meshes. The

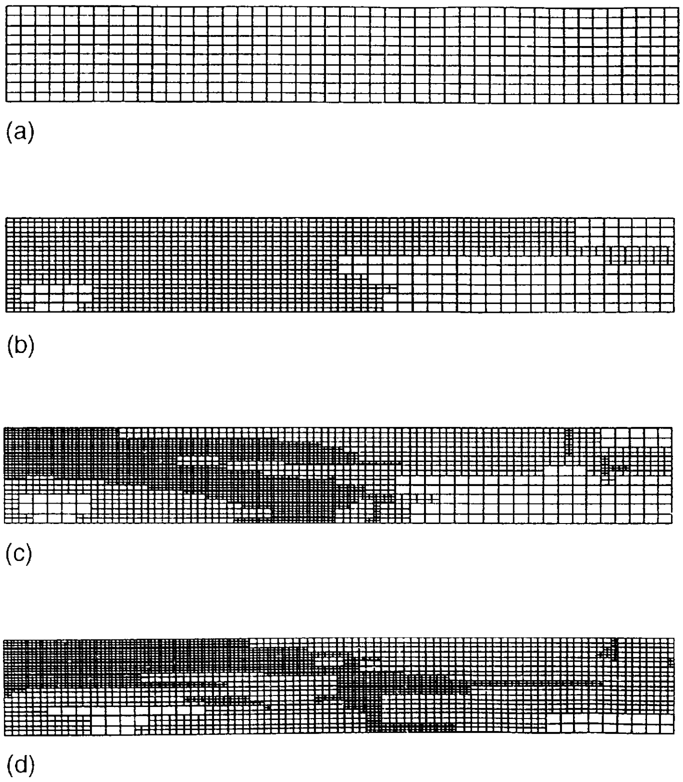

Figure 5. Mesh patterns starting from a regular 100 × 10 mesh, backward-facing step. Only the upstream portion of the mesh is shown for clarity (to $x/h_c = 7$). (a) BFS1U; (b) BFSR1U; (c) BFSR2U; (d) BFSR2C

Table III. Backward-facing step results. Iteration counts and CPU times are normalized by values for the coarsest uniform mesh with upwind differencing. Numbers in parentheses rank the solutions from most accurate (1) to least accurate (6), for upwind and for central differencing independently

Case	$10^9\hat{\dot{K}}_{\mathrm{num,\,g}}$ (equation (4)) (kg m^2 s^{-3})	$10^8\hat{\dot{K}}_{\mathrm{ref,\,1}}$ ($\lvert FLUX\rvert$, equation (9)) (kg m^2 s^{-3})	$10^8\hat{\dot{K}}_{\mathrm{ref,\,2}}$ ($\lvert DISS\rvert$, equation (9)) (kg m^2 s^{-3})	$\overline{IMBAL}_{\mathrm{g1}}$ $\times$ 100 (equation (7)) (%)	$\overline{IMBAL}_{\mathrm{g2}}$ $\times$ 100 (equation (7)) (%)	$\Delta U_{\mathrm{e,g}}$ (equation (10))	Iteration count ratio	CPU time ratio
(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
BFS1U	9·26	3·60	2·68	25·7 (6)	34·6 (6)	0·47 (5)	1·00	1·00
BFS2U	5·10	3·50	2·99	14·6 (4)	17·1 (4)	0·31 (4)	2·87	9·87
BFS3U	2·76	3·43	3·15	8·09 (2)	8·80 (2)	0·16 (2)	8·81	140·2
BFS4U	1·45	3·39	3·24	4·29 (1)	4·48 (1)	0·076 (1)	28·2	3261
BFSR1U	5·62	3·49	2·93	16·1 (5)	19·2 (5)	0·50 (6)	2·57	4·40
BFSR2U	3·98	3·45	3·05	11·6 (3)	13·1 (3)	0·28 (3)	6·28	20·9
BFS1C	2·41	3·33	3·09	7·23 (6)	7·79 (6)	0·22 (6)	1·91	1·93
BFS2C	0·643	3·33	3·27	1·93 (4)	1·97 (4)	0·060 (4)	4·67	15·0
BFS3C	0·167	3·34	3·32	0·50 (2)	0·50 (2)	0·033 (2)	11·1	159·4
BFS4C	0·043	3·34	3·33	0·13 (1)	0·13 (1)	0·00 (1)	44·2	3485
BFSR1C	1·12	3·32	3·21	3·37 (5)	3·48 (5)	0·085 (5)	4·16	7·17
BFSR2C	0·639	3·33	3·26	1·92 (3)	1·96 (3)	0·052 (3)	6·83	27·1

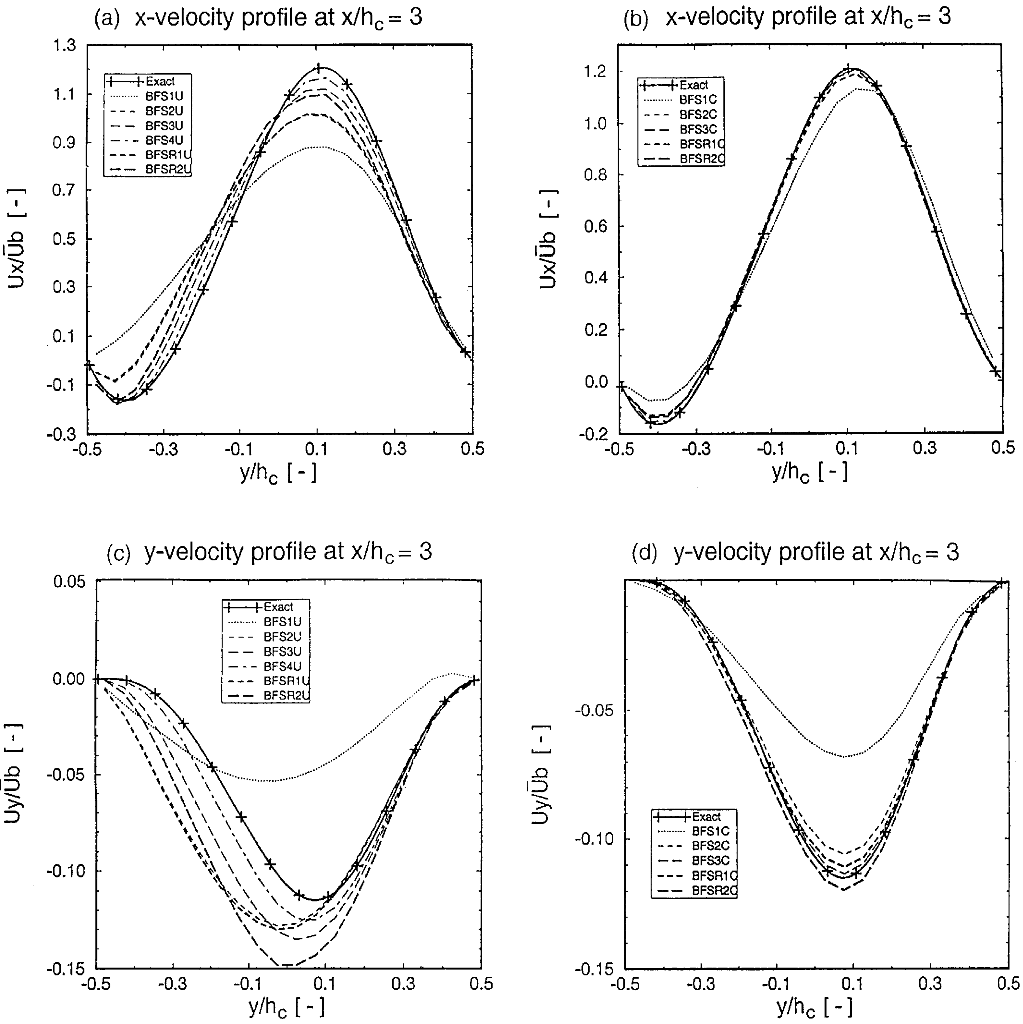

Figure 6. Normalized x- and y-velocity component profiles at $x/h_c = 3$ for backward-facing step: (a) U_x, upwind cases plus benchmark ('exact'); (b) U_x, central cases; (c) U_y upwind cases plus benchmark ('exact'); (d) U_y, central cases

4000-cell non-uniform meshes have global errors that are as much as 25% smaller than those of the 4000-cell uniform meshes, while CPU times for the former are close to double those of the latter. The benefits of local refinement are more apparent with an initially coarse mesh and partially converged solutions (not shown).

7. DISCUSSION

The utility of kinetic-energy-imbalance-based solution-adaptive local refinement has been demonstrated for a shear-driven flow and a pressure driven flow. Key results are threefold. 1) Cell-level imbalances provide a readily computed measure of local solution accuracy from a single-mesh computation. The kinetic energy imbalance emphasizes the energy-containing regions of the flow and is generally consistent with a gradient-based error estimator. 2) The normalized global imbalance provides an indication of overall solution accuracy and a natural criterion for halting refinement. 3) Local solution-adaptive refinement is a cost-effective approach to improved solution accuracy compared with uniform refinement. A combination of local refinement and higher-order differencing appears to be more effective than either strategy by itself. For steady flow problems, initially coarse meshes with refinement based on partially converged solutions further improve efficiency.

An outstanding issue is the correlation between the proposed error estimator and specific engineering quantities of interest. In IC engine combustion, for example, the computed burn rate approaches its asymptotic converged value as the kinetic energy imbalance approaches zero. However, the quantitative correlation between these two quantities cannot be established *a priori*. Also, the correlation will be different for different engineering quantities. For example, the

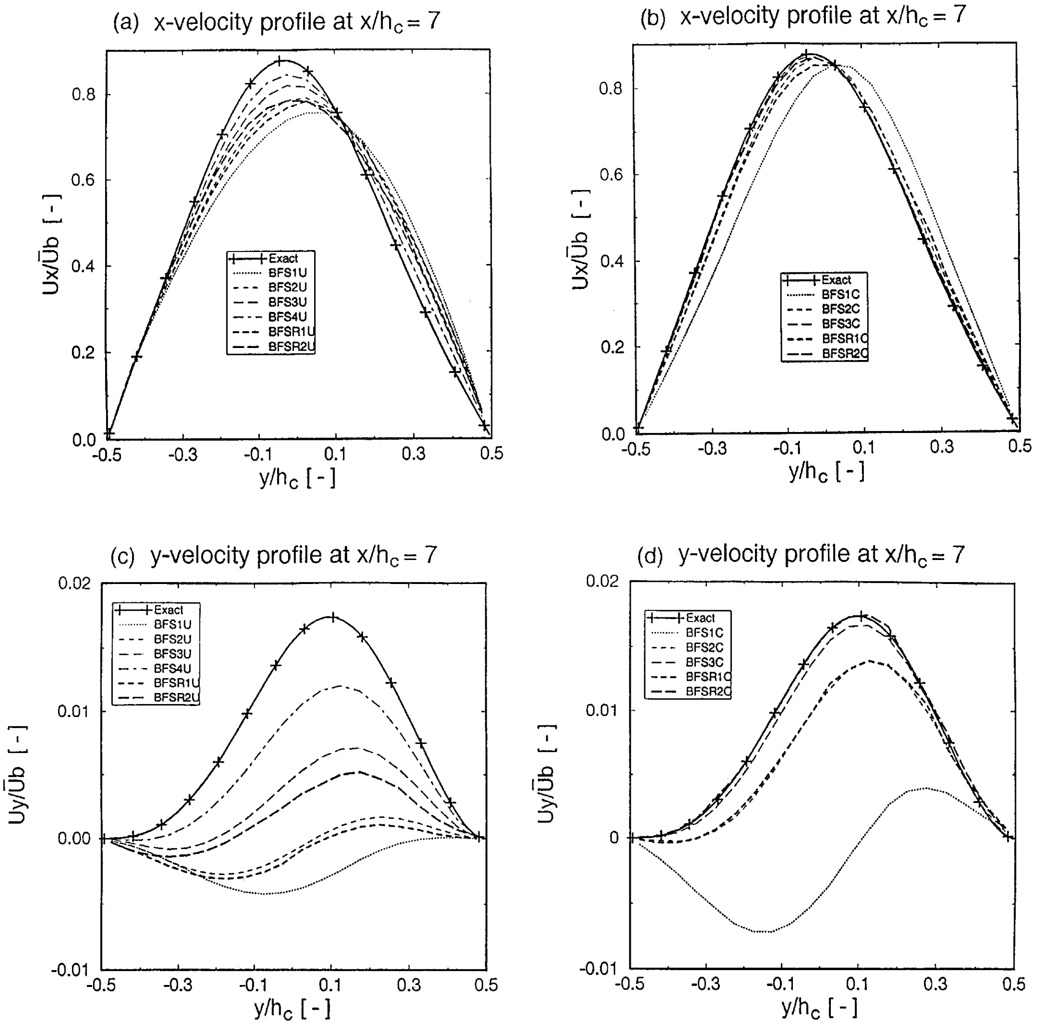

Figure 7. Normalized x- and y-velocity component profiles at $x/h_c = 7$ for the backward-facing step: (a) U_x, upwind cases plus benchmark (‘exact’); (b) U_x, central cases; (c) U_y upwind cases plus benchmark (‘exact’); (d) U_y central cases

quantitative correlation between computed drag coefficient convergence and kinetic energy imbalance in a vehicle aerodynamics application will be different from the burn rate/kinetic energy imbalance correlation of the previous example. The concentration of high-error cells in local mesh regions (e.g. Plate 1(a)) exacerbates the problem of establishing quantitative correlations. This highlights an important advantage of classic grid refinement and the GCI[4] over the current approach: the former can be applied directly to any engineering quantity of interest without the requirement to demonstrate or infer correlations.

On the other hand, classic grid refinement methodology requires a separate estimate of the ‘benchmark’ solution. If the finest grid solution available is adopted as the benchmark, then the order of convergence, p, is obscured, especially for the finer grid cases. Deviation from the benchmark $\Delta U_{\mathrm{e,g}}$ is equal to zero, by definition, for the finest grid solution. Thus n solutions computed on meshes of different densities yield $n-1$ values from which one can verify the convergence order p. Alternatively, the benchmark solution might be estimated using Richardson extrapolation. However, this requires that one lie in the asymptotic range of convergence, which is what one is trying to verify via the grid refinement exercise. By contrast, the benchmark value of the imbalance is known *a priori*: it is identically equal to zero. This distinction can be illustrated using the data of Table III. There the imbalance-based error exhibits the expected asymptotic ratio-of-four decrease with each succesive grid doubling for a second-order method ($p=2$) in the BFS*n*C sequence (column (e): 7·23, 1·93, 0·50, 0·13). The departure-from-the benchmark error sequence for the same problem (column (g)) yields the sequence 0·22, 0·060, 0·033, 0·00: the order of convergence is obscured for the finer meshes as a result of having adopted the finest mesh solution as the benchmark.

No formal proofs of monotonicity of imbalances with mesh refinement or with higher-order spatial discretization are available for the present scheme. Present results and earlier work on realistic three-dimensional time-dependent turbulent flows[3,13] demonstrated monotonic convergence of imbalances in mean kinetic energy, turbulence kinetic energy and angular momentum. In any case the rigour of more formal approaches is lost when one lies outside their asymptotic radius of convergence; this is the regime where one is normally forced to operate in applications.

Several variants of the present strategy can be envisioned. Rather than tagging a specified fraction of cells, f, for refinement (equation (6)), one might find the maximum error over all cells $|\dot{\hat{K}}_{\mathrm{num,\,c\,max}}|$, and refine those cells for which $|\dot{\hat{K}}_{\mathrm{num,\,c}}| > \beta|\dot{\hat{K}}_{\mathrm{num,\,c\,max}}|$, where $0 \leqslant \beta \leqslant 1$. Alternatively, one might split cells only in specified directions using a vector-valued error estimator containing directional information. Efficiency considerations become particularly important in three spatial dimensions. There, refining just 15% of all computational cells doubles the number of cells at each refinement level (equation (5) and (6): $R_{\mathrm{c}} \approx 2$ for $D=3$ and $f=0{\cdot}15$).

Benefits of local refinement are anticipated to be greater in time-dependent problems compared with the present steady flow cases. For the transient port and cylinder with moving piston and valves, for example, the present adaptive meshing strategy is being explored to maintain resolution where it is needed through induction (characterized by steep velocity gradients in annular jets issuing from the valve curtain area), compression (characterized by large variations in cell volume, skewness and aspect ratio in the absence of refinement) and combustion (characterized by steep temperature and concentration gradients). In time-dependent problems, mesh coarsening (local unrefinement) is needed. Unrefinement based on the present kinetic energy imbalance concept has been implemented, with the important restriction that the mesh can become no coarser than the initial mesh; that is, the original parent cells cannot be collapsed. Thus an intelligent choice of initial mesh, with careful consideration of its deformation in moving boundary problems, is imperative. Transient refinement/unrefinement results for benchmark problems including mesh deformation will be the subject of a future paper.

Numerical dissipation of kinetic energy can be of the same order as physical dissipation (viscous dissipation) in complex three-dimensional turbulent flow calculations.[3] In applications including in-cylinder flows, constraints such as the total number of cells or projected CPU time (cost), rather than the desired error level, are likely to limit the level of refinement that is practicable. Rapidly improving computing cost-to-performance ratios coupled with solution-adaptive mesh refinement and higher-order numerical schemes should eventually reduce numerical inaccuracy in three-dimensional time-dependent CFD to the point where the focus can return to physical modelling.

ACKNOWLEDGEMENTS

The authors thank Drs. Sherif El Tahry and Rodney Rask of the GM R&D Center for their support and technical contributions to this project.

REFERENCES

1. S. H. El Tahry and D. C. Haworth, ‘Directions in turbulence modeling for in-cylinder flows in reciprocating engines’, *AIAA J. Propuls. Power*, **8**, 1040–1048 (1992).
2. B. Khalighi, S. H. El Tahry, D. C. Haworth and M. S. Huebler, ‘Computation and measurement of flow and combustion in a four-valve engine with intake variations’, *SAE Paper 950287*, (1995).
3. D. C. Haworth, S. H. El Tahry and M. S. Huebler, ‘A global approach to error estimation and physical diagnostics in multidimensional computational fluid dynamics’, *Int. j. numer. methods fluids*, **17**, 75–97 (1993).
4. P. J. Roache, ‘A method for uniform reporting of grid refinement studies’, in I. Celik, C. J. Chen, P. J. Roache and G. Scheuerer (eds), *Quantification of Uncertainty in Computational Fluid Dynamics*, FED Vol. 158, ASME, New York, 1993, pp. 109–120; ‘Perspective: a method for uniform reporting of grid refinement studies’, *ASME J. Fluids Eng.*, **116**, 405–413 (1994).
5. J. T. Oden, ‘Error estimation and control in computational fluid dynamics’, *Proc. MAFELAP VIII: Mathematics of Finite Elements with Applications*, Uxbridge, April 1993.
6. J. T. Oden, S. R. Kennon, W. W. Tworzydlo, J. M. Bass and C. Berry, ‘Progress on adaptive *hp*-finite element methods for the incompressible Navier–Stokes equations’, *Comput. Mech.*, **11**, 421–432 (1993).
7. R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield and M. L. Welcome, ‘An adaptive Cartesian grid method for unsteady compressible flow in irregular regions’, *J. Comput. Phys.*, **120**, 278–304 (1995).
8. J. Bell, M. Berger, J. Saltzman and M. Welcome, ‘Three-dimensional adaptive mesh refinement for hyperbolic conservation laws’, *SIAM J. Sci. Comput.*, **15**, 127–138 (1994).
9. L. F. Richardson, ‘The deferred approach to the limit’, *Trans. R. Soc. Lond. A*, **226**, 299–361 (1927).
10. M. J. Fritts and J. P. Boris, ‘The Lagrangian solution of transient problems in hydrodynamics using a triangular mesh’, *J. Comput. Phys.*, **31**, 173–215 (1979).
11. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, DC, (1980).
12. P. J. Roache, *Computational Fluid Dynamics*, Hermosa, Albuquerque, NM, 1976.
13. D. C. Haworth, S. H. El Tahry, M. S. Huebler and S. Chang, ‘Multidimensional port-and-cylinder flow calculations for two- and four-valve-per-cylinder engines: influence of intake configuration on flow structure’, *SAE Trans., J. Engines*, (1990); (*SAE Paper 900257*, 1990).
14. U. Ghia, K. N. Ghia, and C. T. Shin, ‘High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method’, *J. Comput. Phys.*, **48**, 387–411 (1982).
15. M. C. Thompson and J. H. Ferziger, ‘An adaptive multigrid technique for the incompressible Navier–Stokes equations’, *J. Comput. Phys.*, **82**, 94–121 (1989).
16. F. Durst and J. C. F. Pereira, ‘Time-dependent laminar backward-facing step flow in a two-dimensional duct’, *J. Fluids Eng.*, **110**, 289–296 (1988).
17. D. K. Gartling, ‘A test problem for outflow boundary conditions—flow over a backward-facing step’, *Int. j. numer. methods fluids*, **11**, 953–967 (1990).
18. P. M. Gresho, D. K. Gartling, J. R. Torczynski, K. A. Cliffe, K. H. Winters, T. J. Garratt, A. Spence and J. W. Goodrich, ‘Is the steady viscous incompressible two-dimensional flow over a backward- facing step at $Re = 800$ stable?’, *Int. j. numer. methods fluids*, **17**, 501–541 (1993).