

**ADAPTIVE FINITE VOLUME METHOD FOR FLOW
PREDICTION USING UNSTRUCTURED MESHES AND
MULTIGRID APPROACH**

by

Samir Muzaferija

Theses submitted for the degree of
Doctor of Philosophy
of the University of London and for the
Diploma of Imperial College

Imperial College of Science, Technology & Medicine

Department of Mechanical Engineering

March - 1994



Abstract

This thesis describes the development and application of adaptive local grid refinement and multigrid methodology for numerical fluid dynamic calculations in complex domains involving body-fitted unstructured meshes. A new discretisation practice was developed, which facilitates space discretisation using cells of arbitrary topology. This practice enables implicit, consistent and uniform treatment throughout the entire computational domain including the interface between refined regions and the rest of the computational mesh.

The integral governing equations employing Cartesian vector and tensor components were discretised in the finite volume manner using a collocated arrangement of variables. The convergence of the SIMPLE segregated algorithm for the solution of the system of coupled nonlinear algebraic equations was accelerated using a 'full approximation storage' multigrid algorithm. A procedure was devised that constructs the hierarchy of numerical meshes used for the multigrid automatically in an adaptive manner. The finer meshes are created from the coarser ones by enriching numerical resolution in regions where the local error indicator was higher than a prescribed value. A new error detection technique was developed from the analysis of the pressure correction term appearing in the cell-face velocity expression for the collocated grid arrangement.

The methodology was assessed by application to two- and three-dimensional problems of engineering interest, for which either benchmark numerical solutions are available or reliable experimental data exist, including laminar two-dimensional driven cavity flow, turbulent flow through an orifice plate, over a surface mounted obstacle and in a gas turbine combustor.

Multigrid acceleration of the iterative solver, in most of the cases, resulted in solution times proportional to the number of grid points, whilst adaptive local mesh refinement allowed uniform resolution of the flow field, with higher density of computational points only in the regions where large gradients or significant numerical diffusion were present.

ACKNOWLEDGEMENTS

I wish to express my deepest gratitude to my supervisor Prof. A. D. Gosman, who has been a great source of advice and guidance for me, and whose continuous interest and support during this study I greatly appreciate.

I am also indebted to my teacher and good friend Prof. I. Demirdzic, from whom I have learned so much about computational continuum mechanics and with whom I have had many stimulating discussions.

I wish to thank all my friends and colleagues in the Prof. Gosman's group, with whom I have shared many pleasant moments and useful discussions, in particular to C. Kralj, H. Weller, B. Adamson and J. Jackson.

Finally, my sincere thanks to Mrs. Nicky Scott-Knight, who has always been interested and concerned with my every-day problems and with whose kind help it was much easier to deal with.

The financial support provided by the Ford Motor Company is gratefully acknowledged.

Dedicated to my parents and sister.

C O N T E N T S

Abstract	I
Acknowledgements	II
Contents	III
Nomenclature	VII

Chapter 1 Introduction

1.1 Background	1
1.2 Previous and Related Studies	5
1.2.1 Multigrid Acceleration Techniques	5
1.2.2 Error Detection Techniques and Adaptive Grid Refinement	8
1.3 Present Contributions	10
1.4 The Thesis Outline	11

Chapter 2 The Governing Equations of Newtonian Flow

2.1 Introduction	13
2.2 General Form of a Conservation Law	14
2.3 Coordinate-Free Form of Governing Equations	16
2.4 Time-Averaged Equations of Turbulent Flow and Turbulence Closure	19
2.5 Boundary Conditions	23
2.6 Closure	24

Chapter 3 Finite Volume Discretisation

3.1 Introduction	25
3.2 Space Discretisation	27
3.3 Discretisation of Equations	31
3.4 Differencing Scheme	34

3.5	Final Form of Discretised Equations	41
3.6	Derivation of Pressure Equation	43
3.6.1	Evaluation of Mass Fluxes	45
3.6.2	Generalised Segregated Approach	48
3.6.3	The SIMPLE Algorithm	50
3.7	Boundary Conditions and their Implementation	52
3.7.1	Inlet Boundaries Definition	52
3.7.2	Outlet Boundaries Definition	53
3.7.3	Symmetry Boundaries Definition	53
3.7.4	Impermeable No-Slip Walls	55
3.8	Solution Procedure for the Coupled Equation Sets	57
3.9	Closure	59

Chapter 4 Multigrid Methodology

4.1	Introduction	61
4.2	Iterative Solution Techniques for Systems of Linear Algebraic Equations	64
4.2.1	Basic Iterative Methods	64
4.2.2	Incomplete LU Factorisation	68
4.2.3	Polynomial Acceleration Techniques	70
4.2.4	Convergence Analysis - Motivation for Multigrid Acceleration	73
4.3	Multigrid Philosophy	79
4.4	Coarse-Grid Equations	82
4.5	Coarse-Grid Boundary Conditions	88
4.6	Components of a Multigrid Solver	89
4.6.1	Coarse and Fine Grid Arrangement	89
4.6.2	Restriction Operator	90
4.6.3	Prolongation Operator	92
4.6.4	Iterative Scheme Used in Inner Iteration Sequence	93
4.6.5	Multigrid Cycle	94
4.7	Closure	97

Chapter 5 Local Grid Refinement and Adaptive Mesh Generation

5.1	Introduction	99
-----	--------------	----

5.2	Numerical Accuracy	102
5.3	Methods for Improving Numerical Accuracy	105
5.4	Local Refinement Indicators	107
5.4.1	Error Estimation Based on Richardson Extrapolation	109
5.4.2	Significance of Pressure-Coupling Term in the Expression for Cell Face Velocity as an Error Estimate for Pressure	110
5.4.3	A New Error Estimation Technique Based on Higher Order Derivatives	112
5.5	Normalisation of the Error Indicator	118
5.6	Overall Algorithm for Multigrid and Local Grid Refinement	120
5.7	Closure	121

Chapter 6 Case Studies

6.1	Introduction	123
6.2	Two-Dimensional Lid Driven Cavity Flow	126
6.2.1	Multigrid Results	128
6.2.2	Multigrid with Adaptive Local Grid Refinement	139
6.2.3	Influence of Use of Different Linear Solvers within the SIMPLE Algorithm on Multigrid Convergence Rate	168
6.3	Flow Through The Orifice Plates	170
6.4	Three-Dimensional Turbulent Flow Over Cubical Obstacle	184
6.5	Three-Dimensional Turbulent Swirling Flow Inside a Model Gas Turbine Combustor	195
6.6	Closure	210

Chapter 7 Summary and Conclusions

7.1	Discretisation	215
7.2	Solution Algorithm and Multigrid Acceleration	217
7.3	Error Estimation and Adaptive Mesh Generation	221
7.4	Future Research	
	References	225

Nomenclature

\mathbf{A}	coefficient matrix
A_m	amplitude factor
a	coefficients of discretised governing equation
$a_{Q\phi}$	implicit part of linearised source term
\mathbf{b}	source vector column of system of algebraic equations
b	source term of discretised governing equation
\underline{b}	external body force vector per unit mass
$b_{Q\phi}$	explicit part of linearised source term
C_j	convective flux through cell face j
$C_{\mu,1,2,3}$	empirical constants (Table 2.1)
c_p	specific heat at constant pressure
$\underline{\underline{D}}$	rate of strain tensor
D_j	diffusive flux through cell face j
\underline{d}	distance vector
d	safety margin used to extend regions marked for refinement
E	explicit part of coefficient matrix \mathbf{A}
E	total energy per unit mass
e	internal energy per unit mass, numerical or solution error
\tilde{e}	numerical error estimate
F	total amount of property f
F_j	mass flux through cell face j
f	conservative property per unit volume
G	rate of production of turbulent kinetic energy
G	matrix used to evaluate gradient of dependent variable
h	characteristic mesh size
$\underline{\underline{I}}$	unit tensor
i	imaginary unit
\underline{i}^i	base vector corresponding to Cartesian direction i
K	implicit part of coefficient matrix \mathbf{A}

K	required number of inner iterations for desired error reduction
k	kinetic energy of turbulence
L	lower triangular matrix
l_{ij}	coefficients of lower triangular matrix
M	iteration matrix
$M\phi$	normalising factor - sum of relevant inflow fluxes
m	wave number
N	number of computational points
$N(\underline{x})$	interpolation function
Pe	Péclet number
p	pressure
Q	source per unit volume
Q_H	heat source per unit volume
Q_w	wall heat flux
\mathbf{q}	heat flux vector
Re	Reynolds number
R_ϕ	normalised sum of absolute residuals
\mathbf{r}	residual column vector
S	surface vector
T	absolute temperature
\mathbf{T}_w	shear surface force at wall
$\underline{\underline{T}}$	stress tensor
T^*	refinement criterion
t	time
U	upper triangular matrix
u_{ij}	coefficients of upper triangular matrix
V	volume
\mathbf{v}	velocity vector
W	computational work
w	weighting factor
\underline{x}	Cartesian position vector
α	under-relaxation factor
γ	blending factor between the CD and UD schemes
Δ	distance between computational points used in 1D analysis
δ_n	normal distance from wall
ε	turbulence dissipation rate
$\varepsilon^{(k)}$	algebraic error column vector at iteration k
ζ	effectivity index (defined by expression (5.3))
κ	von Karman's constant

λ	thermal conductivity
μ	dynamic viscosity
μ_c	convergence factor
v^l	convergence criterion on grid l
ξ	local coordinate in direction connecting two computational points
Φ	exact value of dependent variable
ϕ	approximate value of dependent variable Φ
ϕ'	correction, difference between exact and approximate solution of discretised system of governing equations
ρ	density
$\rho(M)$	spectral radius of matrix M
$\sigma_{k,\varepsilon,t}$	empirical constants (Table 2.1)
τ	time interval, truncation error source or tau error
$\tilde{\tau}$	approximation of tau error
τ^*	normalised tau error
Ω	convergence-rate criterion

Subscripts

c	coarse grid
eff	effective value
f	fine grid
j	index referring to the j^{th} neighbour computational point
max	maximum
min	minimum
P_0	at cell P_0
ref	reference value
S	surface
V	volume
t	turbulent

Superscripts

i	i^{th} Cartesian vector component
(k)	inner iteration counter
T	transpose

X

- ' fluctuating component of dependent variable as a result of time averages
- " fluctuating component of dependent variable as a result of density-weighted averages
- *
- approximate solution of discretised governing equations
- ~ quantities restricted from fine to coarse grid

Mathematical Operators

∇	vector differential operator 'nabla'
\otimes	dyadic product
.	dot product
:	double dot product
$\frac{D}{Dt}$	material or substantial derivative
$\bar{\cdot}$	overbar - time-averaged value
\wedge	density averaged value
(x, y)	scalar product of vectors x and y
I_c^f	restriction operator
I_f^f	prolongation operator

*Chapter 1**Introduction***1.1 BACKGROUND**

Rapid advances in computer hardware in recent years have greatly stimulated the development and application of general numerical algorithms as a credible design aid for complex turbulent flows of industrial and environmental importance. Three-dimensional general-geometry codes coupled with complex turbulence closures can identify qualitative and quantitative trends in highly complex applications, including car bodies, jet-engine combustors, internal combustion engines, complete aircraft and turbine passages. The feasibility of numerical solution procedures is enhanced by the fact that the speed and storage capacity of computers is increasing very rapidly, while their size and cost are falling dramatically.

The Finite Volume (FV) discretisation method, by which the integral formulations of the conservation laws are discretised directly in physical space, gained popularity in the past two decades amongst Computational Fluid Dynamics (CFD) community, and is adopted in the present study. The method takes full advantage of an arbitrary mesh, where a large number of options are open for the definition of a Control Volume (CV) around which the conservation laws are expressed. Because of the nature of FV discretisation, basic quantities (mass, momentum, angular momentum and energy) remain conserved at the discrete level.

The ability of any practical code to yield quantitatively meaningful results depends principally on two factors: the validity of the physical and mathematical models used in describing the flow phenomena; and the accuracy of the numerical approximations of the equations characterising the mathematical model.

Solving the system of the full Navier-Stokes equations is the ultimate goal of a numerical flow simulation. It is accepted, indeed, that all the properties of a continuum fluid flow can be described by the Navier-Stokes equations. Direct solution of these equations in the Reynolds number range where turbulent instabilities occur enables the computation of the stochastic, turbulent fluctuating quantities, out of which Reynolds-averaged mean flow variables can be extracted. However, for practical flows, using current numerical procedures, the computer resources required to resolve the turbulent time and length scales are several orders of magnitude above the computer power expected in the near future, so that turbulence models must still be used. The progress of modelling of turbulence, as well as modelling in general, is paced by limitations to human insight. To improve the comprehension of the physical phenomenon and to assess properly a model describing them, it is important to get numerical solutions based on that model that are free of numerical error.

The present study is concerned with improving the efficiency of numerical solution procedures for complex geometries, relevant to the everyday engineering problems, in order to enable solutions free from numerical error to be achieved economically.

Numerical procedures for solving integral or partial differential equations usually consist of two steps: discretisation of the problem, choosing approximating algebraic equations on a finite-dimensional approximation space, and then constructing a numerical process to solve this, in practice, large system of algebraic equations. The quality of those approximations depends on the complexity of the underlying flow field, the resolution of space and time, and the order of the discretisation. Since the space and equation discretisations are invariably carried out without detailed knowledge of the solution, the discretisation process often

produces a mesh which is not the optimum one. In order to overcome this problem, it has traditionally been necessary to adjust the discretisation by trial and error.

A better solution to this problem is to relate the discretisation to the solution, in an adaptive manner. Using certain criteria, mesh size and/or orders of approximation can be controlled, adapting them to the solution. An initial mesh can be defined to model the basic geometry of the flow domain and the initial and boundary conditions. Thereafter, the strategy is to automatically add or eliminate control volumes (grid points) and/or to change the order of discretisation as needed to alter the local accuracy.

The advantages of using adaptive meshing are twofold. Firstly, the work required to achieve convergence is reduced; and secondly, the computer memory required to obtain a solution of a given accuracy is considerably smaller.

Once the governing equations are approximated with a system of algebraic equations, they have to be solved. The computer work required by most conventional iterative techniques to solve the resulting highly coupled, non-linear system of algebraic equations increases as N^m , where N is the number of computational points and m is of the order 2-3, so that the solutions of very large systems (very fine mesh resolution) can be unacceptably expensive.

Multigrid (MG) methods (Brandt [1977]) take advantage of the fact that the algebraic system is an approximation to the continuous equations and it can itself be similarly approximated by a much smaller algebraic system. This technique has several attractive attributes including linear variation of computing times with N , which is the optimum performance. This is achieved by solving the system of discrete equations on a given grid, by constant interaction with a hierarchy of coarser grids to efficiently annihilate residuals over the whole spatial frequency range, taking advantage of the relations between the different discretisations of the same continuous problem.

There are two broad categories of MG, namely Correction Schemes (CS) and Full Approximation Schemes (FAS). The former operate with corrections to the solution on fine grids, which are eventually added to the fine-grid solution. This method is suitable for linear systems. In contrast, the latter group is better suited to coupled systems of strongly nonlinear algebraic equations by operating on the solution (rather than its correction) on all grid levels. Clearly, it is the FAS which is the appropriate method to adopt in the case of the nonlinear equations of fluid dynamics.

Generally, there is no particular difficulty in applying the MG to a single transport equation nor to a system of linear transport equations discretised on a regular grid. However, a considerable challenge is presented by the highly coupled and complex sets of governing equations describing practical laminar or even more so turbulent flow, especially on an irregular grid. The precise details of the MG implementation are closely associated with the fundamental algorithm used to couple the various sets of algebraic equations, each pertaining to the related flow variable. Most schemes for turbulent recirculating flows solve sequentially the sets of algebraic equations arising from the discretisation of the governing equations, because of the simplicity, flexibility and low-storage requirements. In such circumstances, the resulting solution convergence rate can be seriously hindered if the coupling of the governing equations is not properly honoured. The performance of a MG method depends crucially on the manner in which the MG is integrated into an overall iterative scheme.

Multigrid processing and adaptive discretisation can be used independently. However, their combination is potentially very fruitful (Brandt [1977]), since the MG iterative character fits well into the adaptive process, and these two ideas use and relate similar concepts and similar data structures.

There is strong need to explore the application of adaptive local grid refinement and promising multigrid convergence-acceleration techniques for geometrically and physically complex and practical conditions. The use of unstructured meshes of cells of arbitrary topology facilitates accurate meshing of complex geometries and the use of local, solution-adaptive

mesh refinement. These were primary reasons for adopting this meshing strategy in the present study.

1.2 PREVIOUS AND RELATED STUDIES

1.2.1 Multigrid Acceleration Techniques

The multigrid idea was first suggested by an early work of Fodorenko [1962]. The first practical multigrid algorithms were developed by Brandt [1977], who also suggested its combination with local grid refinement. Fundamental issues and the underlying mathematical theory were further investigated by Hackbusch [1981] and Stuben & Trottenberg [1982]. Much work has been done in the 1980's on extending this technique to the Euler and Navier-Stokes equations, mainly for laminar flow. Ghia et al. [1982] used the stream function-vorticity formulation of the Navier-Stokes equations and employed the strongly-coupled implicit technique of Rubin and Khosla [1981]. They solved the well known driven cavity problem at Reynolds numbers as high as 10000 on a 257x257 grid in approximately 20 to 100 equivalent fine grid iterations and in about a quarter of the computing time required by previous workers. Lacroix et al. [1984] extended an MG variant, also for the stream function-vorticity formulation, to curvilinear coordinates. They pointed out that MG performance can depend heavily on the marching direction of relaxation sweeps within line-implicit algorithms, and recommended the use of ADI-relaxation for strongly recirculating flow. However, the finest grid they employed in a calculation for laminar flow in a curved channel was only 61x21, which in retrospect seems too coarse to permit general conclusions to be drawn on the efficiency of the MG.

The attractiveness of the stream function-vorticity approach derives mainly from the fact that, in 2D flows, the number of equations requiring simultaneous solution reduces from three, when primitive variables are used, to two. However, algorithms operating with primitive variables of velocity and pressure have become dominant and are now widely used,

since there are considerable difficulties encountered when extending the stream function-vorticity formulation to three-dimensional conditions.

Extracting the pressure from the continuity equation for incompressible flow has always been difficult for the primitive variable approach, since the continuity equation, which is left to govern the pressure, does not contain this quantity explicitly. Hence, an algorithm is required to couple the momentum and continuity equations in a manner permitting extraction of the pressure. A significant number of such coupling schemes have been proposed, the most widely used one being the SIMPLE method of Patankar and Spalding [1972].

Within an uncoupled or segregated algorithm, the straightforward implementation of MG is to apply it to each equation set separately, without taking any account of inter-set coupling. Such a simple approach was adopted, for example, by Phillips et al. [1985] in a calculation of a turbulent recirculating flow behind a backward-facing step with the k - ε eddy-viscosity model. In this case, only a very modest saving in CPU time was achieved, due to the uncoupled nature of their MG implementation.

Vanka [1986], recognising the importance of inter-variable coupling to the MG effectiveness, achieved the coupled solution of the momentum and continuity equations implicitly via a symmetrically-coupled Gauss-Seidel (SCGS) smoother. He reported very impressive solution times of up to two orders of magnitude faster than the SIMPLE procedure. Gaskell et al. [1987] compared the SCGS used on its own, the SCGS combined with the MG and the SIMPLE method used with a single grid only. They computed the same cavity flows considered by Vanka. The SCGS without multigrid took twice the time as the traditional SIMPLE strategy. However, the SCGS with multigrid was 45 times faster than the SIMPLE without multigrid.

Comparison of the results for MG efficiency arising from the works by Phillips et al. [1985] and those of Vanka [1986] and Gaskell et al. [1987] reveals that inter-variable coupling within the MG algorithm is of crucial importance, and in some way must be introduced into segregated pressure-correction algorithms, if the MG strategy is to yield significant benefits.

Sivaloganathan and Shaw [1988] recognised this need and chose the SIMPLE algorithm to be the smoother embedded in their FAS-MG. They reported computations for laminar lid-driven-cavity flows at Reynolds numbers between 1 and 10000, and demonstrated that the convergence rate was independent of mesh resolution. Sivaloganathan and Shaw's algorithm used a staggered finite-volume arrangement: although this offers a high level of stability in the context of pressure-correction schemes, it necessitates tedious interpolation practices within the restriction and prolongation operations.

Solution schemes employing colocated (non-staggered) arrangements have been increasingly adopted over the past decade, following the pioneering work of Rhie and Chow [1983], who developed a special interpolation practice, which effectively introduced artificial fourth-order pressure diffusion, suppressing odd-even oscillations in the pressure field. The colocated grid arrangement is highly advantageous in conjunction with MG, due to its simplicity enabling easy mesh coarsening and refining.

Barcus et al. [1987] were the first to combine the CS-MG with the cell-centred colocated finite volume discretisation and the SIMPLE solution algorithm as the relaxation scheme within a Full Multigrid (FMG) procedure. The MG implementation was similar to that adopted by Sivaloganathan and Shaw, in that it accounts for inter-variable coupling. The results presented for the laminar lid driven cavity flow and backward facing step followed well the multigrid theory developed for linear equations. Becker et al. [1988] extended the method of Barcus et al. to FAS-MG, in which approximate solutions are used as coarse grid variables instead of corrections as in CS-method. They exploited another important feature of FAS, which is straightforward extension to local grid refinement.

Turbulent flow has not been much studied. Following a rather discouraging foray with the MG into turbulence by Phillips et al. [1985]. Peric et al. [1989] reported attaining speed-up factors of the order of 30 when applying FAS-FMG with grids of up to 20480 nodes to a turbulent backward-facing step flow. The colocated finite-volume scheme incorporated a high-Re number $k-\varepsilon$ eddy-viscosity model with wall

functions accounting for the viscous near-wall processes. The authors pointed out, however, that the wall-function approach become questionable with increasing grid refinement, as the wall-adjacent grid nodes could shift into the viscous sublayer, where the log-law was no longer valid. More recently Lien & Leschziner [1991] and Lien [1992] performed wider investigations into multigrid convergence acceleration for complex two and three dimensional turbulent flows. Three turbulence models were considered: low-Re and high Re variants of the two-equation $k-\varepsilon$ eddy-viscosity model, and a Reynolds stress transport closure. Considerable performance variations have been observed, the most influential factor being geometry and the flow type. The CPU savings in turbulent flow were significant, but far less dramatic than in some laminar flows. This was ascribed to the addition of a highly non-linear turbulence-transport model to the coupled equation set, but also a result of strongly increased grid non-uniformity and convective processes.

1.2.2 Error Detection Techniques and Adaptive Grid Refinement

Applications of local grid refinement to fluid dynamics calculations, using finite volume discretisation, are rare, despite its potential for reducing both computer time and storage requirements. High Reynolds or Mach number flows generally have thin regions of substantial velocity variation, such as boundary layers and shock waves. Those regions require much finer mesh spacing than most of the domain. For such problems, adaptive grid techniques are much more efficient than uniform fine-grid solution techniques. As mentioned above, multigrid methods entail the use of grids of different density, and it is therefore particularly convenient to combine these methods with local grid refinement.

Phillips & Schmidt [1984] reproduced the accuracy of a 33x33 grid solution using a locally-refined mesh with 30% of cells and 40% of the computational work, for a two-dimensional diffusion problem. Thompson & Ferziger [1989] presented an automatic adaptive refinement technique coupled to the multigrid approach for solving the steady-state Navier-Stokes equations. The adaptive grid approach was adopted from the work of Caruso [1985] and based on estimation of the local truncation error, as this was the source of solution error. Second-order finite volume

approximations to the Navier-Stokes equations and a deferred correction scheme were used, enabling second-order accurate solutions on a staggered grid. Lagrange interpolation with four, nine and sixteen points, were used for both prolongation and restriction: these tests indicated that it is especially bad to use high-order interpolation for the coarse to fine grid transfer. The adaptive refinement approach reduced the computer memory and CPU time to 20 and 40 % of the requirements of the "pure" multigrid method applied to the lid driven cavity flow.

A very similar implementation of local grid refinement was done by Becker et al. [1988] but using the colocated grid arrangement. The calculation was done for a backward facing step at various Reynolds numbers. Although the CG and FMG were sensitive to the aspect ratio of the grid control volume, the combination of FAS and local grid refinement showed marked improvements in convergence rates and computing times relative to the single-grid strategy. Smith [1990] used the MG algorithm with the SIVA algorithm of Caretto et al. [1972a] adopted for colocated grids. His results for two-dimensional lid driven cavity flow were similar of those reported by Thompson & Ferziger [1989].

All these implementations of local grid refinement have used structured, orthogonal grids, where there was a need for special treatment at interfaces between locally-refined patches and the rest of the numerical mesh. These interfaces were treated in an explicit manner as internal boundaries. Such a treatment might slow-down the convergence, especially if the locally refined mesh is made of many patches such that this interface is large.

The use of MG methods combined with unstructured meshes and finite volume discretisation of Navier-Stokes equation has not been explored extensively so far. A study by Mavriplis [1988] applied multigrid to unstructured triangular meshes to solve the two-dimensional Euler equations. Since his method assumes no relation between the various meshes, a tree-search algorithm was used to identify more efficiently the regions of overlap between coarse and fine grid cells, so that variables could be transferred back and forth between them. His method was applied to both cell-centred and cell-vertex discretisations of the Euler equations,

and results for flows around multi-element airfoils were presented. For both discretisation schemes, convergence was accelerated by an order of magnitude.

In summary, all efforts reviewed above provide encouraging evidence that the MG method performs well even in physically complex conditions, including turbulent flow, and that its combination with adaptive local grid refinement is in many ways beneficial, primarily in considerably reducing memory demands, while retaining the desired level of accuracy.

1.3 PRESENT CONTRIBUTIONS

The present study makes the following specific contributions to the field of computational fluid dynamics:

- A new discretisation technique, which facilitates the use of arbitrary computational cell topology and is performed in real space using Cartesian coordinates, has been developed for application to the governing equations in integral form. The technique does not assume that the value of a variable prevails in the whole cell, but rather assigns this value to a particular point in the control volume. The variation of the dependant variable is determined by a shape function, whose coefficients are determined by fitting the values of the dependant variable in the neighbourhood of the control volume. The resulting over-determined system of liner algebraic equations, which is a consequence of the fact that the number of cell neighbours is arbitrary, is solved by a minimisation technique. This gives flexibility to the scheme, demonstrated by the fact that one of its formulations presented in this study is second-order accurate on a non-uniform grid. In addition the discretisation practice is easy to understand and implement.
- The generalised segregated approach of Van Doormaal and Raithby [1985] has been applied to non-orthogonal meshes and in the light of the present discretisation technique.

- The significance of the pressure-coupling term, appearing in the cell-face velocity expression for the colocated grid arrangement (Rhie & Chow [1983]), as an error detection measured is demonstrated and used to estimate the error in approximating the variation of pressure.
- A novel error estimation technique, similar to that used by McGuirk and Rodi [1978] and McGuirk et al. [1980], is developed and used to assess the numerical error introduced by discretising convective and diffusive transport. The methodology is such that its implementation to unstructured meshes made of cells of arbitrary topology is simple and computationally inexpensive.
- The developed discretisation practice facilitates implicit, consistent and uniform treatment throughout the entire computational domain, including the interface between refined regions and the rest of the computational mesh.
- Using the hierarchy of computational grids obtained during the process of adaptive local grid refinement, a multigrid accelerator has been built into the solution algorithm, capable of handling three-dimensional complex laminar and turbulent flows, and unstructured meshes.
- The presented methodology, involving multigrid combined with local grid refinement, has been successfully applied to turbulent three-dimensional flows, including the flow over a prismatic obstacle and the swirling flow inside a gas-turbine combustor.

1.4 THE THESIS OUTLINE

In the following the control volume concept has been introduced by Reynolds' transport theorem, and the time-averaged transport equations for mass, momentum, energy, kinetic energy of turbulence and its dissipation are summarised in coordinate-free form.

Chapter 3 describes a conservative finite volume discretisation technique as a possible way of solving the set of conservation equations assembled in Chapter 2. This involves both space discretisation and the equation discretisation. The successful colocated grid methodology of Rhie and Chow [1983] selected for use here is outlined and the treatment of non-linearity and pressure-velocity coupling is examined. The final form of the discretised equations is presented, along with the steps required to obtain a solution.

In Chapter 4 the multigrid technique for solving a set of simultaneous algebraic equations is described. Some well established iterative methods are briefly reviewed, and the reason for the inefficiency of the Gauss-Seidel scheme is illustrated. The multigrid method for overcoming this inefficiency is then introduced. A basic multigrid algorithm is outlined, and the main multigrid components are discussed.

In Chapter 5, numerical accuracy is examined, and methods for improving it are discussed. Error indicators for identifying regions requiring local refinement and the basics of adaptive discretisation are described.

In Chapter 6, the performance of the present solution procedure is evaluated by application to two- and three-dimensional problems of engineering interest, for which either benchmark numerical solutions are available or reliable experimental data exist, including laminar two-dimensional driven cavity flow, turbulent flow through an orifice plate, over a surface mounted obstacle and in a gas turbine combustor.

Finally, in Chapter 7, the thesis is summarised, and the main conclusions and suggestions for future research are given.

Chapter 2

The Governing Equations of Newtonian Flow

2.1 INTRODUCTION

The laws of fluid dynamics were established from observation that a certain number of physical system quantities are conserved. Formal derivation of the governing equations of fluid dynamics follows from Reynolds' transport theorem, which facilitates implementation of laws and principles concerning the behaviour of a system (material or Lagrangian volume composed of the same particles) in an arbitrary control volume. Introduction of the control volume concept and Reynolds' transport theorem leads in a logical way to a finite volume method and conservative discretisation, which will be presented in Chapter 3.

The laws of conservation of mass, linear and angular momentum and energy completely determine the behaviour of the system and are independent of the nature of the fluid. However, they do not form a closed system of equations and additional *subsidiary laws or constitutive relations* that apply to specific fluids are necessary for their closure. For various reasons it is often not practically possible to describe completely the evolution of the fluid flow, and it is necessary to define mathematical models which reduce the complexity of the original basic equations and make them tractable within limits of computer technology. The derivation of the governing equations, their closure and definition of appropriate boundary condition, which together form a well-posed mathematical problem, will be the subject of this chapter.

First, in Section 2.2 Reynolds' transport theorem will be briefly described, as the starting point for the control volume concept. In Section 2.3 the coordinate-free form of the governing differential equations will be given, and in Section 2.4 the time-averaged equations of turbulent flow and the k - ε model of turbulence for their closure will be presented. In Section 2.5 the mathematical description of fluid flow will be closed by boundary conditions. Finally, the conclusions will be summarised in Section 2.6.

2.2 GENERAL FORM OF A CONSERVATION LAW

Reynolds' transport theorem (Slattery [1981], Aris [1989]) concerns the rate of change of the volume integral of any continuous function of space coordinates and time $f(\underline{x},t)$ within an arbitrary control volume V , bounded by a closed surface S (Figure 2.1), where \underline{x} is a position vector and t is time. The total amount of the property f in the volume V is a function of t that can be calculated as:

$$F(t) = \int_V f(\underline{x},t) dV \quad (2.1)$$

It can be shown that *the material derivative* of F (the time rate of increase of F within a system which occupies the volume V at time t):

$$\frac{DF}{Dt} = \frac{\partial F}{\partial t} + \underline{v} \cdot \nabla F \quad (2.2)$$

is equal to the integral over the volume V of the rate of change of f at a point plus the net flow of f through the surface S , thus:

$$\frac{DF}{Dt} = \int_V \frac{\partial f}{\partial t} dV + \int_S f \underline{v} \cdot \underline{dS} \quad (2.3)$$

where $\underline{v}(x,t)$ is the velocity vector and $d\underline{S}$ is the outward-pointing surface-element vector. Equation (2.3) is used in converting laws from the system form (Lagrangian method of analysis) to the Control Volume (CV) form (Eulerian method of analysis).

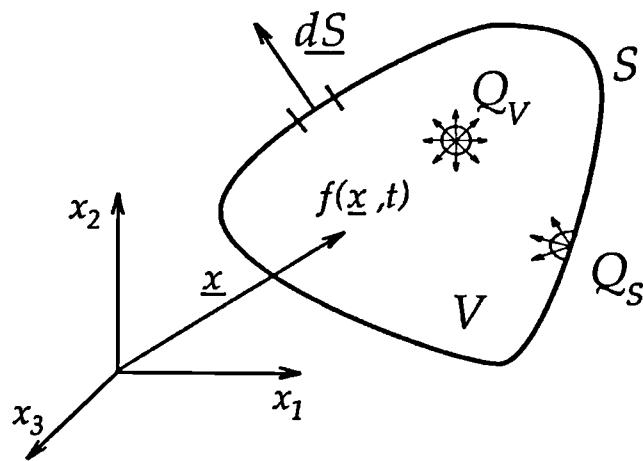


Figure 2.1 Control volume

The time rate of increase of F within a system caused by volume $Q_v(x,t)$ and surface $Q_s(x,t)$ sources is:

$$\frac{DF}{Dt} = \int_V Q_v dV + \int_S Q_s \cdot d\underline{S} \quad (2.4)$$

Substituting expression (2.4) into (2.3), one gets the general conservation equation (Scipio [1967]) in integral form:

$$\int_V \frac{\partial f}{\partial t} dV + \int_S (f \underline{v} - Q_s) \cdot d\underline{S} = \int_V Q_v dV \quad (2.5)$$

The consequence of conservation is that the variation of a conserved flow quantity f within a given control volume is due to the net effect of some internal sources and of the amount of the quantity that is crossing

the boundary surface. The latter is called the flux. The fluxes and sources are, in general, dependent on the space-time coordinates as well as on the fluid motion. Finally it should be noted that quantity f can be any scalar, vector or tensor field.

2.3 COORDINATE-FREE FORM OF GOVERNING EQUATIONS

In this section the general conservation equation (2.5) will be applied to the conservation of mass, linear momentum, angular momentum and energy conservation, which completely determine the behaviour of a fluid flow which can be described in terms of basic dependent variables (velocity, pressure, density and temperature fields).

The mass conservation expresses the empirical fact that, if there are no mass sources or sinks in the system, mass cannot disappear from the system nor be created. Considering F to be the total mass of the volume V , then f is the fluid density ρ , and if the total mass of the volume is to be conserved, then the source terms Q_s and Q_v must vanish, and the general conservation equation (2.5) reduces to:

$$\int_V \frac{\partial \rho}{\partial t} dV + \int_S \rho \underline{v} \cdot \underline{dS} = 0 \quad (2.6)$$

This is called the continuity equation.

The conservation of linear momentum asserts that the sum of body and surface forces equals the time rate of change of linear momentum. In the general conservation equation (2.5), which now expresses the balance of momentum, F is the total momentum of the volume V , f the momentum per unit volume $\rho \underline{v}$, Q_v body force $\rho \underline{b}$ and Q_s is the stress tensor $\underline{\underline{T}}$:

$$\int_V \frac{\partial(\rho \underline{v})}{\partial t} dV + \int_S (\rho \underline{v} \otimes \underline{v} - \underline{\underline{T}}) \cdot d\underline{S} = \int_V \rho b dV \quad (2.7)$$

Equation (2.7) is known as *Cauchy's first law of motion* and holds for any continuum. For a Newtonian (linear Stokesian) fluid in local thermodynamic equilibrium, the stress tensor $\underline{\underline{T}}$ is equal to:

$$\underline{\underline{T}} = -(p + \frac{2}{3} \mu \nabla \cdot \underline{v}) \underline{\underline{I}} + 2\mu \underline{\underline{D}} \quad (2.8)$$

where p is the pressure, μ the dynamic viscosity of the fluid, $\underline{\underline{I}}$ is the unit tensor and $\underline{\underline{D}}$ is the rate of strain tensor, defined as:

$$\underline{\underline{D}} = \frac{1}{2} [\nabla \otimes \underline{v} + (\nabla \otimes \underline{v})^T] \quad (2.9)$$

The conservation of angular momentum asserts that the time rate of change of angular momentum is equal to the moment of the resultant force acting on the system. When the linear momentum is balanced, then the balance of angular momentum requires that the stress tensor be symmetric (Aris [1989]):

$$\underline{\underline{T}}^T = \underline{\underline{T}} \quad (2.10)$$

if there are no internal angular momentum charges arising from couple stresses and body couples. This is *Cauchy's second law of motion*. Since the stress tensor (2.8) is symmetric, the conservation of angular momentum in a Newtonian fluid is always satisfied.

The conservation of energy expresses the empirical fact that the time rate of change of the total energy E (internal plus kinetic as well as others like chemical, nuclear etc. which will be neglected in this study) is equal to the rate of work done on the system or portion thereof plus the heat transmitted to the system. Considering the general conservation equation for the quantity E , f stands for the total energy of the medium per unit volume ρE , consisting of its internal energy ρe and kinetic energy $\frac{1}{2} \rho \underline{v} \cdot \underline{v}$:

$$\rho E = \rho(e + \frac{1}{2} \underline{v} \cdot \underline{v}) \quad (2.11)$$

The surface source Q_s is the result of the work done on the fluid by the internal stresses acting on the surfaces $\underline{T} \cdot \underline{v}$ and the heat flux \underline{q} , crossing the surface of the control volume. Q_v is the volume source, which is the sum of the work of the body forces $\rho \underline{b} \cdot \underline{v}$ and the volume heat source Q_H (radiation, chemical reactions etc.). Substituting all these terms in the general conservation equation (2.5), the energy conservation equation in integral form becomes:

$$\int_V \frac{\partial(\rho E)}{\partial t} dV + \int_S [\rho E \underline{v} - (\underline{q} + \underline{T} \cdot \underline{v})] \cdot d\underline{S} = \int_V (\rho \underline{b} \cdot \underline{v} + Q_H) dV \quad (2.12)$$

The heat flux \underline{q} is the diffusion flux vector, defined as the heat transport present in a medium due to molecular thermal agitation, which obeys Fourier's law of heat conduction:

$$\underline{q} = \lambda \nabla T \quad (2.13)$$

where the proportionality coefficient λ is the thermal conductivity coefficient and T is the absolute temperature.

Equation (2.12) is an equation of change for the sum of e and $\frac{1}{2} \underline{v} \cdot \underline{v}$. By combining this equation with equation (2.7) one obtains an equation of change for e :

$$\int_V \frac{\partial(\rho e)}{\partial t} dV + \int_S (\rho e \underline{v} - \underline{q}) \cdot d\underline{S} = \int_V (\underline{T} : \nabla e \underline{v} + q_H) dV \quad (2.14)$$

Equation (2.14) is called the *equation of thermal energy* and it is generally more convenient for heat-transfer analysis than the equation of total energy.

The thermodynamic laws define the internal energy e as a function of two other thermodynamic variables, chosen from pressure p , density ρ , temperature T , or any other intensive variable, for example:

$$e = e(p, T) \quad (2.15)$$

Finally, in order to close the system of governing equations, an equation of state is required, the form of which is in general:

$$\rho = \rho(p, T) \quad (2.16)$$

and in case of incompressible flow asserts that ρ is constant.

2.4 TIME-AVERAGED EQUATIONS OF TURBULENT FLOW AND TURBULENCE CLOSURE

Most of the flow situations occurring in the nature enter into a particular *state of continuous instability*, called *turbulence*, and can be said to be *steady* on an average basis only, since small scale high frequency statistical fluctuations of all the flow quantities, in both space and time, are always present (Launder & Spalding [1974]). A flow exhibiting these macroscopic fluctuations is called *turbulent flow*. Otherwise, a well-ordered flow, free of macroscopic velocity fluctuations is called *laminar*. The transition from laminar to turbulent flow depends on the *Reynolds number*, which is the ratio of the inertia forces to the viscous forces.

Numerical simulation of the large and small-scale turbulent motion can be done by solving the full three-dimensional time-dependent governing equations of viscous flow (Section 2.3), on a mesh with spacing smaller than the length scale of smallest turbulent eddies, and with a time step smaller than the smallest time scale of turbulent fluctuations. Thus the numerical description of the turbulent fluctuations is a task which puts very high demands on computer resources. The alternative is to use a statistical description of turbulent motion, formulated in terms of

averaged quantities. One such description, adopted in the present study is time-averaging (Tennekes & Lumley [1989]), whereby each dependent variable is expressed as the sum of its *mean*, or time-averaged value $\bar{\Phi}$, and *fluctuating* component Φ' :

$$\Phi = \bar{\Phi} + \Phi' \quad (2.17)$$

where:

$$\bar{\Phi}(\underline{x}, t) = \frac{1}{\tau} \int_{-\tau/2}^{\tau/2} \Phi(\underline{x}, t + \xi) d\xi \quad (2.18)$$

and the time interval τ is large enough with respect to the time scale of the turbulent fluctuations, but small with respect to the scale of other time-dependent effects. The separation (2.17) is known as *Reynolds decomposition*. For compressible flows the averaging process leads to products of fluctuations between density and other variables such as velocity or internal energy. In order to avoid their explicit occurrence a *density-weighted* average is introduced (Cebeci & Smith [1974]), through:

$$\hat{\Phi} = \frac{\overline{\rho \Phi}}{\overline{\rho}} \quad (2.19)$$

with

$$\Phi = \hat{\Phi} + \Phi'' \quad (2.20)$$

and

$$\overline{\rho \Phi''} = 0 \quad (2.21)$$

This way of defining mean turbulent variables removes all products of density fluctuations with other fluctuating quantities. Applied to the governing equations of the previous section, the following equations for the turbulent mean mass, momentum and energy are obtained:

$$\int_V \frac{\partial \bar{\rho}}{\partial t} dV + \int_S \hat{\rho} \hat{\underline{v}} \cdot \underline{dS} = 0 \quad (2.22)$$

$$\int_V \frac{\partial \bar{\rho} \hat{v}}{\partial t} dV + \int_S (\bar{\rho} \hat{v} \hat{v} - (\underline{T} - \overline{\rho \underline{v}'' \otimes \underline{v}''})) \cdot \underline{dS} = \int_V \bar{\rho} \underline{b} dV \quad (2.23)$$

$$\int_V \frac{\partial \bar{\rho} \hat{e}}{\partial t} dV + \int_S (\bar{\rho} \hat{e} \hat{v} - (q - \overline{\rho e'' \underline{v}''})) \cdot \underline{dS} = \int_V (\underline{T} : \nabla \underline{v} + q_H) dV \quad (2.24)$$

In these expressions, all variables are considered as averaged quantities (density and pressure as time averages and velocity and internal energy as density-weighted averages). The time-averaging procedure produces the *Reynolds stress* term $\overline{\rho \underline{v}'' \otimes \underline{v}''}$ and the turbulent heat diffusion vector $\overline{\rho e'' \underline{v}''}$, in the momentum and energy equations respectively. Since these quantities are unknown, in particular their relationship with the mean flow dependent variables, the use of the time-averaged equations for computation of turbulent flows requires modelling of these unknown relations.

The time-averaged conservation equations are often closed by the often used $k-\epsilon$ model of turbulence (Jones & Launder [1972]) for Reynolds stresses and the turbulent heat flux vector. It is adopted in this investigation.

The $k-\epsilon$ model is based on the Boussinesq assumption (Boussinesq [1877]), which assumes that the turbulent Reynolds stress and scalar fluxes are linked to the time averaged flow properties in a fashion analogous to their laminar flow counterparts. The model can be summarised as follows. In equation (2.8) and (2.13) μ , λ and p are replaced by:

$$\mu_{eff} = \mu + \mu_t \quad (2.25)$$

$$\lambda_{eff} = \lambda + \lambda_t \quad (2.26)$$

and

$$P = p + \frac{2}{3} (\mu_t \nabla \cdot \underline{v} + \rho k) \quad (2.27)$$

where μ_t and λ_t are the turbulent eddy viscosity and diffusivity coefficients defined as:

$$\mu_t = C_\mu \rho \frac{k^2}{\varepsilon} \quad (2.28)$$

$$\lambda_t = \frac{\mu_t}{\sigma_t} \quad (2.29)$$

Here, k stands for the kinetic energy of turbulence and ε for the turbulence dissipation rate. They are defined as

$$k = \frac{1}{2} \overline{(\underline{v}' \cdot \underline{v}')} \quad (2.30)$$

$$\varepsilon = \frac{1}{\rho} \overline{\underline{T}' : (\nabla \otimes \underline{v}')} \quad (2.31)$$

and are obtained by the solution of their respective transport equations (Launder & Spalding [1974]), of the same form as the general conservation equation (2.5):

$$\int_V \frac{\partial \rho k}{\partial t} dV + \int_S (\rho k \underline{v} - \mu_k \nabla k) \cdot \underline{dS} = \int_V (G - \frac{2}{3} (\rho k + \mu_t \nabla \cdot \underline{v}) \nabla \cdot \underline{v} - \rho \varepsilon) dV$$

(2.32)

and

$$\int_V \frac{\partial \rho \varepsilon}{\partial t} dV + \int_S (\rho \varepsilon \underline{v} - \mu_\varepsilon \nabla \varepsilon) \cdot \underline{dS} = \int_V (C_1 \frac{\varepsilon}{k} G - C_2 \rho \frac{\varepsilon^2}{k} - C_3 \rho \varepsilon \nabla \cdot \underline{v}) dV$$

(2.33)

The eddy diffusivities μ_k and μ_ε associated with the kinetic energy and dissipation equations respectively are defined by:

$$\mu_k = \mu + \frac{\mu_t}{\sigma_k} \quad \text{and} \quad \mu_\varepsilon = \mu + \frac{\mu_t}{\sigma_\varepsilon} \quad (2.34)$$

The rate of production of turbulent kinetic energy G is modelled as:

$$G = 2\mu_t \underline{D} : \nabla \underline{\underline{v}} \quad (2.35)$$

The quantities $C_\mu, C_1, C_2, C_3, \sigma_k, \sigma_\epsilon$ and σ_t are empirical coefficients and the values used in this study were taken from Launder & Spalding [1974], and are given in Table (2.1).

C_μ	C_1	C_2	C_3	σ_k	σ_ϵ	σ_t
0.09	1.44	1.92	-0.33	1.0	1.2	0.9

Table 2.1 The values of empirical coefficients in the $k-\epsilon$ model of turbulence

Introducing the $k-\epsilon$ model into the time-averaged governing equations leads to a system that is formally identical to the instantaneous one (equations 2.6-2.16), where the molecular viscosity μ and the thermal conductivity λ are replaced by $\mu+\mu_t$ and $\lambda+\lambda_t$. In addition, the turbulent kinetic energy contribution to the normal stress is included in the mean pressure P according to equation (2.27).

2.5 BOUNDARY CONDITIONS

A mathematical model is *well-posed* (Garabedian [1964]) if the solution exists, is unique and depends in a continuous way on boundary conditions. The last condition means that a small perturbation of boundary conditions should give rise to a small variation of the solution at any point of the domain at a finite distance from the boundaries. Boundary conditions are conventionally divided into two groups:

- i) Dirichlet conditions when the values of dependent variables on the boundaries of the flow domain are specified, e.g.

$$\Phi = F_s \quad \text{on } S \quad (2.36)$$

ii) Neumann conditions when the gradient of dependent variables on the boundaries of the flow domain is imposed, e.g.

$$\nabla \Phi = \underline{F}_s \quad \text{on } S \quad (2.37)$$

where F_s and \underline{F}_s are given scalar and vector functions respectively, defined on the surface $S(\underline{x})$, which bounds the domain considered.

2.6 CLOSURE

In this chapter the control volume concept has been introduced with Reynolds' transport theorem, and the integral form of governing equations, expressing the concept of conservation of certain properties, is given in coordinate-free form. The time-averaged transport equations for mass, momentum, energy, kinetic energy of turbulence and its dissipation with the boundary conditions appropriate to the given flow constitute the mathematical model which is to be solved. Having established the mathematical framework, the discretisation of space and the transport equations is the next step in achieving the numerical solution. This is the subject of the following chapter.

Chapter 3

Finite Volume Discretisation

3.1 INTRODUCTION

In the previous chapter the general mathematical model of Newtonian fluid flow has been given. In this study, only steady flow ($\Phi=\Phi(\underline{x})$) is considered. The momentum equation, which describes the conservation of a vector quantity (linear momentum), is resolved in Cartesian spatially invariant coordinate directions x_i ($i=1,2,3$), which results in three equations in terms of velocity vector components v_i . In this case the conservation of a vector means conservation of its components too, which facilitates a fully conservative integration of the governing equations.

The next step in achieving a final numerical solution is the choice of the discretisation method, which involves two components: the *space discretisation* and the *equation discretisation*. The space discretisation consists of defining a numerical grid, which replaces the continuous space by a finite number of computational points where the numerical values of the dependent variables will be determined. Replacement of individual terms in the governing equations by algebraic expressions connecting nodal values on a finite grid is called equation discretisation.

The basis of all numerical methods consists of this transformation of the governing equations into an algebraic system of equations, with subsets of these approximating each conservation equation. This chapter describes the *finite volume* method (FV) which is adopted in the present study. The present method takes full advantage of an arbitrary mesh and,

by the direct discretisation of the integral form of the conservation laws, ensures that the basic quantities (mass, momentum, energy) will also be conserved at the discrete level. This is the most fundamental property of this numerical scheme.

The nonlinearity and interequation coupling of the original conservation equations reappear in the algebraic equations through coefficients that depend on the solution to the equations, and through coupling between the subsets of the equations. In order to resolve the nonlinearity, *outer iterations* are set up for which the coefficients of the nonlinear system are treated as constants and updated every iteration, according to the current available estimates of dependent variables. The iterations of iterative methods for the solution of linear algebraic system are called *inner iterations*. The terms of the system of algebraic equations are said to be treated *implicitly* if they are updated during inner iterations. The terms kept constant during inner iterations and calculated using the values of dependent variables from previous outer iteration are said to be treated *explicitly*.

An algorithm can be built of outer iterations to handle interequation coupling, where the subsets of linear algebraic equations representing the equations of mass, momentum and scalar quantities (energy, k , ε) conservation, are solved sequentially. In the case of incompressible or low Mach number flows, the effects of pressure on density are negligible. The continuity equation is an additional constraint on the velocity field, which can be satisfied, together with momentum conservation, through the pressure field. Combining momentum and mass conservation equation in discretised form, the continuity equation is transformed into a pressure equation which has the same general form as the other discretised equations. Details of these steps will be given in the remainder of the present chapter.

First, in Section 3.2 the computational grid, which is the numerical representation of the fluid flow domain, is defined. The general conservation equation (2.5) is discretised on this grid in the finite volume manner in Section 3.3, while in Section 3.4 the details of discretisation are described. In Section 3.5 the final form of discretised equations is

presented. Section 3.6 presents the calculation of mass fluxes for a colocated grid arrangement, and a generalised segregated approach for pressure-velocity coupling with details of the SIMPLE (Patankar & Spalding [1972]) algorithm. The numerical implementation of boundary conditions used in the present study is presented in Section 3.7. Finally, the sequence of operations that makes up a typical outer iteration cycle for the solution of coupled systems of non-linear algebraic equations is presented in Section 3.8, followed by the closing remarks given in Section 3.9.

3.2 SPACE DISCRETISATION

The governing equations of Section 2.3 completely determine the behaviour of a fluid flow in a solution domain. This behaviour can be described by space and time distribution of basic dependent variables within that domain. If one assumes certain variations of dependent variables in space and time, the governing equations may be integrated, and if it happens that the assumed profiles are exact, these equations will be satisfied identically together with prescribed boundary conditions. However, in general this is not the case. In order to make these assumptions less important, the solution domain is subdivided into a *finite* number of controls volumes (CV) or cells, which are fitted to follow the shape of the domain, such that even a very simple (i.e. linear) assumptions of variable variations within those control volumes will produce plausible results. In the present study each control volume is defined by a set of faces, each face by a set of vertices, and each vertex is defined by its position vector with respect to a global Cartesian coordinate system. The computational points are located in the centre of the control volumes ('the cell centred' arrangement), and all variables share the same control volumes ('the colocated' or 'non-staggered' arrangement).

The generation of meshes for complex geometries is a problem the importance of which increases with the space dimension, making this aspect one of the most important in the calculation of three-dimensional

flow in complex domains. A review of available and popular techniques for the numerical grid generation in CFD is presented in the book of Sengupta et al. [1988].

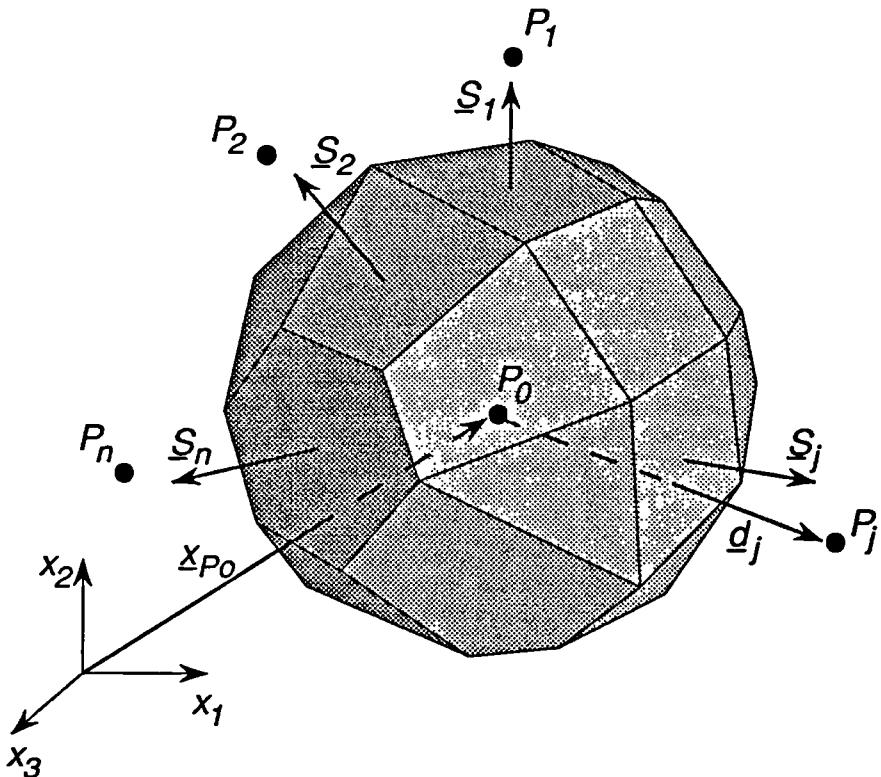


Figure 3.1 A general polyhedral control volume

Meshes made of polyhedral cells, bounded by cell faces S_j (Figure 3.1), and in general of different topology, combined in an arbitrary way (Figure 3.2) are used for the space discretisation in the present study. This, the most general approach to mesh generation, offers flexibility in meshing complex geometrical configurations when accurate description of boundaries is desired and facilitates efficient local grid refinement, which can be carried out only in the regions where refinement is necessary, without disturbing the rest of the computational mesh. The later issue is of particular interest to the present study, in which one of the major concerns is adaptive local grid refinement. The price to be paid for the beneficial features of *unstructured* meshing using cells of arbitrary topology in comparisons to *structured meshes*, (Hirsch [1991a]) is

additional bookkeeping as well as the necessity of devising a discretisation strategy which is capable of dealing with different cell topologies.

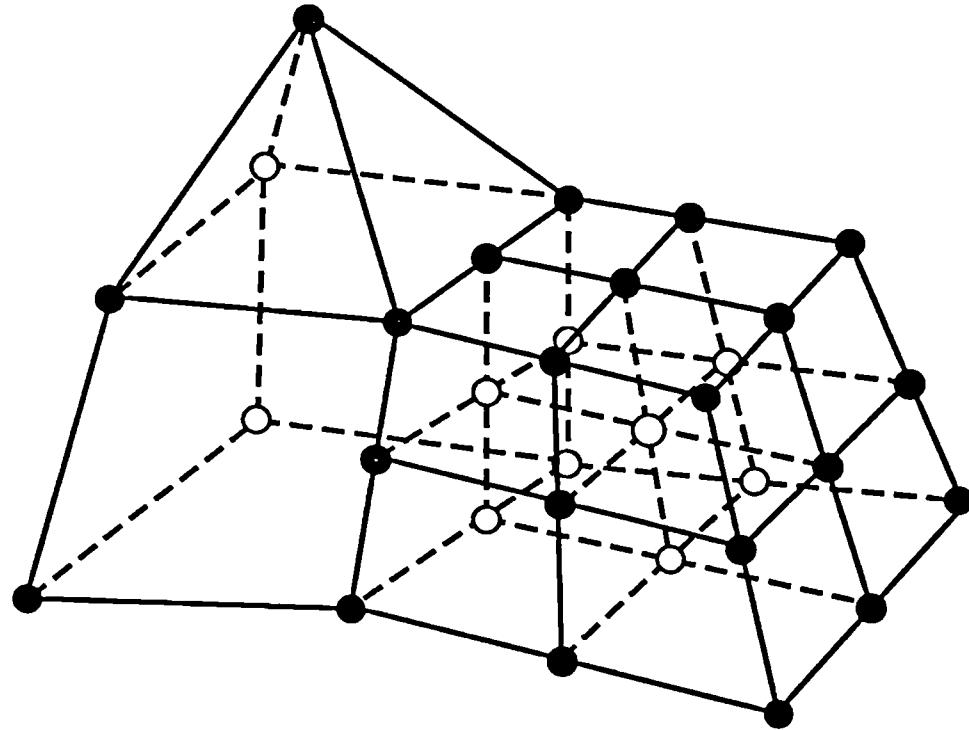


Figure 3.2 Unstructured mesh composed of cells of different topology

For the discretisation practice described in the subsequent sections the information about the numerical mesh, which should be provided by a pre-processor, can be divided into the geometrical data related to cell centres, the geometrical data related to cell faces, and the data that defines the connectivity between the cell faces and the cells. The geometrical cell centre data is made of the list of the cell volumes V_{Pj} and the cell centre coordinates \underline{x}_{Pj} (Figure 3.1). The geometrical cell face data contains the list of the surface vectors \underline{S}_j and the coordinates of the cell face centres \underline{x}_j . The cell face connectivity data contains for each face the indices of the cells which share that face. It is important to note that all this information can be extracted from the vertex definition of the faces and face definition of the cells.

Since the spatial variation of dependent variables is unknown, certain assumptions of their variation in space have to be made. If a discretisation technique is numerically convergent (Fletcher [1991]), then the error introduced by those assumptions is directly dependent on a characteristic size of the considered volume, and in the limit when the control volume shrinks to a computational point the exact solution of governing equations will be obtained, and the only remaining error will be that due to modelling of turbulence. However, the finer the grid, the larger the number of computational points and increased computer time and memory, both of which are limited by available computational facilities.

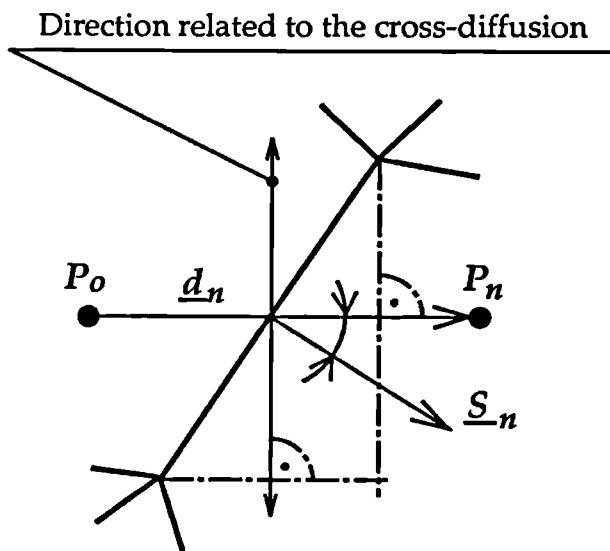


Figure 3.3 Mesh non-orthogonality and cross-diffusion

The accuracy of the numerical solutions, the stability and the rate of convergence of the solution procedure, all depend to a certain degree on particular properties of the computational grid. The desirable properties of the numerical mesh, with respect to the discretisation practice described in the following sections, are:

- The angle between the surface vector of the cell face \underline{S}_n and the vector \underline{d}_n (Figure 3.3) that connects two computational points of cells that share that cell face, should be as small as possible. This will reduce diffusive

fluxes in the direction normal to \underline{d}_n (the so called *cross-diffusion* flux, see Figure 3.3) improving the convergence and stability of a numerical solution scheme.

- The concentration of computational points (control volumes) should be higher in regions of stronger spatial variation of dependent variables. This property is discussed in Chapter 5.

3.3 DISCRETISATION OF EQUATIONS

The steady-state forms of the conservation equations of Sections (2.3) and (2.4) can be cast in the following form when applied to a control volume V_{P_o} , (Figure 3.1) :

$$\sum_{j=1}^n \int_{S_j} (\rho \underline{v} \Phi - \lambda_\Phi \nabla \Phi) \cdot \underline{d}S = \int_{V_{P_o}} Q_\Phi dV \quad (3.1)$$

<i>Convection</i>	<i>Diffusion</i>	<i>Source</i>
-------------------	------------------	---------------

where n is the number of neighbour cells that share cell faces with cell P_o . In the case of the momentum equation (2.23) resolved in Cartesian coordinate directions x_i , the diffusion process is not completely described by a gradient-type law such as (2.13), and all terms that do not fit into gradient expression of equation (3.1) are to be included in the source term Q_Φ .

Equation (3.1) relates the total source of Φ in the control volume with the fluxes of Φ crossing the control volume surfaces by diffusion and convection. This equation is exact. Approximations are introduced when the local distributions of Φ and Q_Φ over the control volume are assumed in order to obtain an algebraic equation of the form:

$$a_0\phi_{P_0} - \sum_{j=1}^n a_j\phi_{P_j} = b_\phi \quad (3.2)$$

The lowercase ϕ in the equation (3.2) is introduced because the solution to these equations at the grid points will be in general different from the exact solution Φ , but as noted earlier will become an increasingly good approximation to the latter as the control volume dimensions are reduced. The form of approximation of $\Phi(\underline{x})$ is discussed in the next section. At this point it will be assumed that the field of variable Φ in a neighbourhood of point P_0 may be approximated by the following function (Zienkiewicz [1977], Hirsch [1991a]):

$$\phi(\underline{x}) = \sum_{j=0}^n \phi_{P_j} N_j(\underline{x}) \quad (3.3)$$

where $N_j(\underline{x})$ are the interpolation functions.

Equation (3.1) has three distinctive parts: convection, diffusion and source. The source term is integrated over the cell volume, whereas the convection and diffusion terms form the sum of fluxes through the CV faces. The flux through the cell face j has to be calculated uniquely, i.e. it has to have the same intensity and opposite signs for cells P_0 and P_j . This essential property has to be satisfied by the numerical discretisation of the flux contributions in order for a scheme to be conservative (Roache [1972], Gosman & Lai [1982]). The evaluation of each term of equation (3.1) will now be described.

Convection term C_j gives the rate at which variable Φ enters or leaves the control volume through the cell face S_j due to the mass flux F_j :

$$F_j = \int_{S_j} \rho \underline{v} \cdot d\underline{S} = (\rho \underline{v})_j \cdot S_j \quad (3.4)$$

The velocity \underline{v}_j at the CV face, needed for the evaluation of the mass fluxes, is obtained from a special interpolation practice which assures a stable solution procedure with the colocated grid arrangement as described later

in Section 3.6.1. The convection flux of the variable Φ through the CV face j may now be approximated as follows:

$$\boxed{C_j = \int_{S_j} \rho v \Phi \cdot d\underline{S} \approx F_j \phi_j} \quad (3.5)$$

where ϕ_j stands for the cell face mean value of the variable Φ . To obtain an approximation of its value expression (3.3) may be used, substituting the position vector \underline{x} by the coordinates of the cell face centre \underline{x}_j .

Diffusion term D_j of Φ through the cell face j is dependent on the gradient of variable Φ at the face. Expression (3.3) may be used to find an approximation of the gradient of Φ at the cell centre:

$$\nabla \phi_{P_0} = \sum_{j=0}^n \phi_{P_j} \nabla N_j(\underline{x}) \quad (3.6)$$

If this gradient is then interpolated between two cells which share the face j

$$\nabla \phi_j = w_{P_0} \nabla \phi_{P_0} + w_{P_j} \nabla \phi_{P_j} \quad (3.7)$$

then an approximation of $\nabla \Phi$ at \underline{x}_j will be obtained. The coefficients w_{P_0} and w_{P_j} are weighting factors. Using these concepts the diffusion term can be approximated by:

$$\boxed{D_j = - \int_{S_j} \lambda_\Phi \nabla \Phi \cdot d\underline{S} \approx -\lambda_{\phi_j} \nabla \phi_j \cdot \underline{S}_j} \quad (3.8)$$

where λ_{ϕ_j} stands for the cell face mean value of the diffusivity, obtained by suitable interpolation between cells which share the cell face j .

The source term is obtained by integrating the specific source Q_Φ over the control volume P_0 :

$$\int_{V_{P_0}} Q_\Phi dV = Q_\Phi^{P_0} V_{P_0} \quad (3.9)$$

where $Q_\Phi^{P_0}$ is the volume V_{P_0} mean value of the specific source Q_Φ .

Assembling the contributions from convective and diffusive fluxes through all faces and integrating the source term over all control volumes comprised in the computational domain results in a system of N algebraic equations of the form (3.2), which link, for each control volume, the value of the dependent variable at the control volume centre with the values at the points in its neighbourhood. N is the total number of control volumes involved in space discretisation of the domain.

3.4 DIFFERENCING SCHEME

In the previous section the discretisation of the model transport equation in the finite volume manner was performed. To do this it was necessary to assume certain variations of variable Φ in the neighbourhood of point P_0 in order to evaluate an approximation ϕ , and its gradient at the cell face j (equation 3.3). In principle, the choice of interpolation assumptions for the variation of the dependent variables will not affect the final solution, provided sufficiently fine grids are employed and the adopted discretisation practice is numerically convergent, as described in Section 3.2. For multidimensional flow phenomena, the capacity and speed of present-day computers generally proves to be the limiting factor in the use of very fine grids. Therefore, the true merit of the higher-order schemes is that they approximate better the spatial variation of the dependent variable, which is in general nonlinear, so they may be sufficiently accurate to permit the performance of complex calculations using presently available computing resources. Both the stability of the numerical algorithm and the accuracy of the solution obtained depend on the type of interpolation practice (differencing scheme) used to estimate the value of variable Φ and its gradient at the cell face in expressions (3.5)

and (3.8). The properties of some differencing schemes are investigated here and more about them may be found in e.g. Patel et al. [1987] and Hirsch [1991a, 1991b].

The majority of the existing finite volume CFD codes have adopted the finite-difference approach for evaluating the value of the dependent variable and its gradient at the cell face (e.g. TEACH of Gosman & Ideriah [1983], KIVA of Amsden et al. [1985]), by assuming a certain variation of variable Φ along the coordinate lines of a local coordinate system connecting neighbouring points with point P_o , which are preferably mutually orthogonal. The task becomes more complicated, if one decides to use curvilinear coordinate systems, when extra curvature terms start to appear. Discretisation of those terms introduces extra computational cost and can be a potential source of numerical errors (Demirdzic [1982]). The use of cells of different topology, when independent directions of the local coordinate system are no longer well-defined (e.g. triangular cells in two-dimensional discretisation) makes that the whole method lose its generality and special treatment is required.

The alternative to the finite difference approach is one close to the concept of finite elements. In this, the distribution of variable Φ is sought in the space surrounding point P_o rather than along certain lines in the space. This is the path followed here.

In the present study, a linear variation of variable Φ is assumed i.e.:

$$\phi(\underline{x}) = \phi_{P_o} + (\nabla\phi)_{P_o} \cdot (\underline{x} - \underline{x}_{P_o}) \quad (3.10)$$

where \underline{x}_{P_o} is the position vector of point P_o and $(\nabla\phi)_{P_o}$ is an approximation of the gradient of ϕ at point P_o . It is quite clear that in practice any other profile could be assumed. The unknown three components of $(\nabla\phi)_{P_o}$ are to be determined by demanding that the profile (3.10) fits the values of Φ at chosen locations. The reason that a linear profile is used in the present study is that the unknown components of $(\nabla\phi)_{P_o}$ may be determined by using the values of the dependent variable at the nearest neighbours of cell P_o only, since for the most simple cell

topology (tetrahedral), the number of nearest neighbours (four) is sufficient to determine $(\nabla\phi)_{P_0}$. The fact that only nearest neighbours are used results in a compact computational molecule. However, an attempt to fit the values of Φ at all nearest neighbours results in an overdetermined algebraic system. In the present work a least-squares fit of (3.10) to the set of the nearest neighbour values is proposed to calculate $(\nabla\phi)_{P_0}$, i.e.

$$\frac{\partial}{\partial(\nabla\phi)_{P_0}^i} \sum_{j=1}^n (\phi_{P_j} - \phi(\underline{x}_{P_j}))^2 = 0 \quad i=1, 2, 3 \quad (3.11)$$

where $\phi(\underline{x}_{P_j})$ is the value of function (3.10) at \underline{x}_{P_j} , ϕ_{P_j} is the actual value of ϕ at P_j and $(\nabla\phi)_{P_0}^i$ is the i^{th} Cartesian component of the vector $(\nabla\phi)_{P_0}$ with respect to the frame of reference. The calculation of gradient based on the least-square method is valid for an arbitrary cell topology.

In terms of one-dimensional Taylor Series Truncation Error (TSTE) analysis, the gradient obtained by equation (3.11) is second-order accurate on a uniform grid only. In order to achieve an approximation of second order on a non-uniform grid, than the influence of the distance vector modulus $|\underline{d}_j| = |\underline{x}_{P_j} - \underline{x}_{P_0}|$ (Figure 3.1) has to be taken into account:

$$\frac{\partial}{\partial(\nabla\phi)_{P_0}^i} \sum_{j=1}^n \frac{1}{|\underline{d}_j|} \left(\frac{\phi_{P_j} - \phi_{P_0}}{|\underline{d}_j|} - (\nabla\phi)_{P_0} \cdot \frac{\underline{d}_j}{|\underline{d}_j|} \right)^2 = 0 \quad i=1, 2, 3 \quad (3.12)$$

Expression (3.12) is equivalent to the requirement that the projection of $(\nabla\phi)_{P_0}$ onto \underline{d}_j fits the gradient based on the values of variable ϕ in that direction and the influence of point P_j on the gradient is weighted by the modulus of the distance vector.

The solution of the system of algebraic equations (3.12) can be expressed in matrix form as:

$$(\nabla\phi)_{P_0} = G^{-1} h \quad (3.13)$$

where h and the coefficients g_{kl} of the 3×3 matrix G are defined by:

$$\boxed{h_k = \sum_{j=1}^n \frac{\phi_{P_j} - \phi_{P_o}}{|\underline{d}_j|} \frac{d_j^k}{|\underline{d}_j|^2}}$$

$$g_{kl} = \sum_{j=1}^n \frac{d_j^k d_j^l}{|\underline{d}_j|^3} \quad (3.14)$$

where n is the number of nearest neighbours and d_j^k is the k^{th} Cartesian component of the vector \underline{d}_j . It is noteworthy that the matrix G is symmetric, that its coefficients depend on the cell geometry only, and it is therefore the same for all dependent variables. If there is no mesh motion, it is enough to calculate G^{-1} once and to store its six independent coefficients for each control volume.

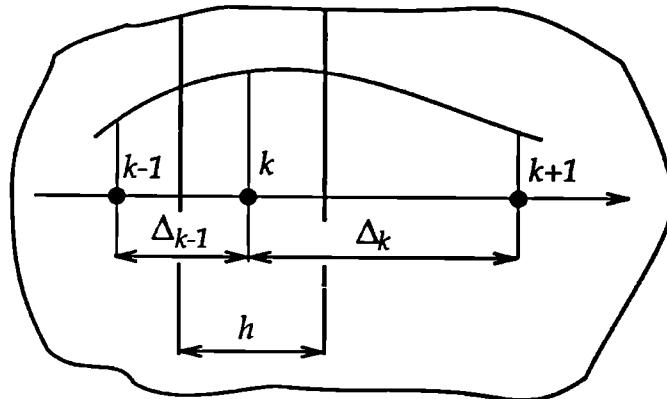


Figure 3.4 One dimensional finite-volume grid

In the one-dimensional case (Figure 3.4) expression (3.13) gives:

$$\nabla \phi_k = \frac{\Delta_{k-1}}{\Delta_{k-1} + \Delta_k} \frac{\phi_{k+1} - \phi_k}{\Delta_k} + \frac{\Delta_k}{\Delta_{k-1} + \Delta_k} \frac{\phi_k - \phi_{k-1}}{\Delta_{k-1}} \quad (3.15)$$

An expression like (3.15) can also be derived from a Taylor series expansion analysis, where the leading term in the truncation error

expression $\nabla^3 \Phi_k \frac{\Delta_{k-1} \Delta_k}{6}$ goes to zero as the second power in Δ_k , i.e. the scheme is second-order. For the constant distance between computational points $k-1$ and $k+1$ this term has its maximum value for $\Delta_{k-1} = \Delta_k$.

Once the assumption about the distribution of the dependent variable is made, one can start evaluating diffusive and convective fluxes. However, at this stage one has to be aware of the numerical implementation which can, if not carried out with due caution, cause unphysical results and, in some cases, numerical instability. Attention will be paid to this where necessary in what follows.

The diffusive flux of the conserved variable Φ may now be calculated using expressions (3.8) and (3.13). However, the gradient of the transported variable obtained by (3.13) is second-order space centred, and as such, cannot sense oscillations in the solution with period equal to twice that of the characteristic mesh size h (see Section 3.6.1). Therefore direct implementation of the gradient obtained from (3.13) could cause decoupling, with the result that, once generated, nonphysical oscillations would remain in the final solution. This can be corrected by the explicit addition of recoupling terms (Caughey & Jameson [1982]), achieved in the present study, by replacing the gradient $\nabla \phi_j$ resulting from (3.7) with $\nabla \phi_j^*$:

$$\nabla \phi_j^* = \nabla \phi_j - \left(\omega \nabla \phi \cdot \frac{\underline{d}_j}{|\underline{d}_j|} + (1-\omega) \frac{\phi_{Pj} - \phi_{Po}}{|\underline{d}_j|} \right) \frac{\underline{d}_j}{|\underline{d}_j|} + \frac{\phi_{Pj} - \phi_{Po}}{|\underline{d}_j|} \frac{\underline{d}_j}{|\underline{d}_j|} \quad (3.16)$$

where ω is a blending factor with values between 0 and 1. Through expression (3.16) a fraction of the component of the gradient $\nabla \phi_j$ in direction \underline{d}_j is replaced by the gradient explicitly calculated from the value of the dependent variable difference across that cell face. It should be noticed that the component of the gradient calculated explicitly across the face is second order accurate on uniform mesh only. In the present work ω is chosen to be 1, what simplifies the expression (3.16) reducing the computational effort.

The convective flux (3.5) depends on the values of dependent variables at the cell face, affecting strongly both the stability of numerical algorithm

and the accuracy of the solution obtained. The properties of differencing schemes used to model convection have been extensively investigated and more about them may be found in e.g. Patel et al. [1987] and Hirsch [1991a]. Here, attention will be concentrated on the second-order central space discretising scheme (CDS) and the first-order upwind differencing scheme (UDS), which are both used in the present study, because of their beneficial features and easy implementation.

The Central Differencing Scheme (CDS) assumes a linear spatial variation of dependent variable, as has been assumed by expression (3.10), which in the one dimensional case (Figure 3.5) results in:

$$\phi_j = \phi_{P_0} + \frac{\phi_{P_j} - \phi_{P_0}}{x_{P_j} - x_{P_0}} (x_j - x_{P_0}) \quad (3.17a)$$

or in general:

$$\phi_j = \phi_{P_0} + \nabla\phi_j (x_j - x_{P_0}) \quad (3.17b)$$

In terms of the TSTE the CDS is second-order accurate (Lai [1982]), but it may generate non-physical oscillations (Section 5.2) in regions where Taylor series expansion analysis is not valid, (e.g. too coarse grid or sharp discontinuities in ϕ). In addition, this scheme gives for Péclet number (the ratio of the convective flux to the diffusive flux) $Pe > 2$ a negativ

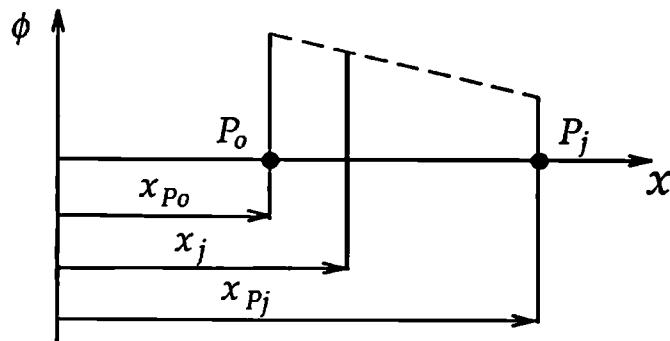


Figure 3.5 Schematic representation of the CDS

contribution to the coefficients of the discretised system (3.2) which can violate the Scarborough criterion (Patankar [1980]), and have undesirable

effects on iterative solution procedures. A more detailed discussion about this will appear in the next chapter.

The Upwind Differencing Scheme (UDS) also known as the 'donor cell' scheme (Gosman et al. [1969]) is directed towards introducing physical properties of the flow equations into the discretised formulation. The convected cell face value is taken to be that at the upstream node:

$$\phi_j = \begin{cases} \phi_{P_0} & \text{if } F_j \geq 0 \\ \phi_{P_j} & \text{if } F_j < 0 \end{cases} \quad (3.18)$$

In terms of the TSTE the UDS is first-order accurate and therefore has truncation error proportional to a second derivative, which acts as an additional *numerical (false) diffusivity* (Raithby [1976], Patankar [1980]), which dampens the high-frequency components of the solution, and smoothes out strong gradients (Section 5.2). The upwind scheme contribution to the coefficients is always non-negative. This is accomplished by sacrificing accuracy.

Many workers have opted for the upwind scheme in preference to higher order schemes, such as CDS, for the already mentioned reasons. One way around the iterative instability of the CDS is the *deferred-correction* procedure first suggested by Khosla & Rubin [1974]. In this, the first-order upwind values are used on the left-hand side of the equation (3.2) and difference between first- and second-order approximations (expression in {} in (3.19)) taken at the previous outer iteration level is used on the right-hand side i.e.:

$$\phi(\underline{x}_j) = \begin{cases} \phi_{P_0} + \{\gamma \nabla \phi_j \cdot (\underline{x}_j - \underline{x}_{P_0})\} & \text{if } F_j \geq 0 \\ \phi_{P_j} + \{\gamma \nabla \phi_j \cdot (\underline{x}_j - \underline{x}_{P_j})\} & \text{if } F_j < 0 \end{cases} \quad (3.19)$$

where γ is a blending factor. At convergence, the solution is second-order accurate and the convergence properties are similar to those of the first-order approximation.

In order to damp non-physical oscillations, some numerical diffusivity can be introduced by blending first- and second-order schemes by appropriately adjusting γ in equation (3.19). In the limits the scheme is second-order (CDS) for $\gamma=1$ and first-order (UDS) for $\gamma=0$. The main problem of those so-called *flux-blending methods* is how to determine the local value of γ for each cell face. One of the first and perhaps most widely used methods employing a form of flux blending technique, is the hybrid scheme of Spalding [1972]. Later Boris & Book [1973], Chapman [1981], Peric [1985] and Ziman [1990] suggested different ways of determining the blending factor, but none of these were general and simple for implementation. Furthermore, the dynamic and discontinuous nature of updating those blending factors sometimes caused instability of the iterative process for the steady-state calculation.

It has been suggested by Gresho & Lee [1981] that instead of suppressing 'wiggles' by using upwind differencing, they should be used to indicate the inadequacies of the computational mesh, which should then be refined accordingly. Brandt & Yavneh [1991] showed that even very simple problems are not approximated well by the upwind difference scheme with anisotropic artificial diffusivity, regardless of numerical parameters, and that the bad discretisation breeds poor multigrid convergence, since coarse grids do not yield proper approximations to smooth fine-grid errors.

In the present study a constant blending factor γ is taken in the whole domain, the value being set on a case by case basis. This blending factor may be large since, in a process of adaptive local grid refinement, we hope to resolve all regions where the solution is not smooth by adding sufficient computational points, such that the CDS will not produce nonphysical oscillations.

3.5 FINAL FORM OF DISCRETISED EQUATIONS

In this section the final form of the coefficients and 'source' of equation (3.2) will be presented. The coefficient a_j has contributions from diffusion

and convection and expresses that part of the influence of the dependent variable value at point P_j on the value at P_o that is treated implicitly. The coefficients a_j and source terms are defined as follows:

$$a_j = \frac{\lambda_{\phi j}}{|\underline{d}_j|^2} (\underline{d}_j \cdot \underline{S}_j) + (\max(F_j, 0) - F_j) \quad (a1) \quad (a2) \quad (3.20)$$

$$a_o = -a_{Q_\phi} + \sum_{j=1}^n a_j \quad (a3) \quad (a4) \quad (3.21)$$

$$b_\phi = \sum_{j=1}^n \lambda_{\phi j} (\nabla \phi_j - (\nabla \phi_j \cdot \frac{\underline{d}_j}{|\underline{d}_j|}) \frac{\underline{d}_j}{|\underline{d}_j|}) \cdot \underline{S}_j + \quad (b1) \quad (3.22)$$

$$+ \sum_{j=1}^n \gamma \nabla \phi_j \cdot ((\underline{x}_j - \underline{x}_{P_o}) \max(F_j, 0) + (\underline{x}_j - \underline{x}_{P_j}) \max(-F_j, 0)) + b_{Q_\phi} \quad (b2) \quad (b3)$$

In the above the component of the gradient in the direction of the vector \underline{d}_j , expressed by (3.16), is treated implicitly (a1), and when the distance vector \underline{d}_j and surface vector \underline{S}_j are collinear then only this component of the gradient causes a diffusive flux. The part of the diffusion flux due to the component of the gradient normal to the distance vector \underline{d}_j (the 'cross-diffusion') vanishes when the grid is orthogonal, and is small compared to the other part if the grid non-orthogonality is not severe. For this reason it is treated explicitly (b1), which means that only the contributions of the nearest neighbours of node P_o are treated implicitly.

The convective flux, according to deferred-correction practice, is split into implicit (a2) and explicit (b2) parts.

The volumetric source term defined by (3.9) is linearised (Patankar [1980]), thus:

$$Q_\phi^{P_0} V_{P_0} = (a_{Q_\phi} \phi_{P_0} + b_{Q_\phi}) V_{P_0} \quad (3.23)$$

$$a_{Q_\phi} \leq 0$$

where a_{Q_ϕ} and b_{Q_ϕ} are coefficients of the linearised source term Q_ϕ . The purpose is to allow the source term to be split into contributions to the central coefficient (a3) and source term (b3) of discretised equation. The condition (3.23) imposed on a_{Q_ϕ} will increase the diagonal dominance of the coefficient matrix, which enhances the convergence of iterative linear algebraic system solvers (Varga [1962]). Finally, the central coefficient a_0 has contributions from convection and diffusion (a4), and is equal to the sum of all neighbouring coefficients a_j ($j=1,2,\dots,n$).

The momentum component equations are a special case, since they contain a surface source term, reproduced below, which cannot be cast completely into gradient form. This term is also treated explicitly and discretised similarly to the discretisation of convective and diffusive terms described above, i.e.

$$\begin{aligned} q_{v_i} &= \sum_{j=1}^n \int_{S_j} -p \underline{i}^i + \mu(\nabla \otimes \underline{v}) \cdot \underline{i}^i \underline{l} \cdot d\underline{S} = \\ &\approx \sum_{j=1}^n [-p_j \underline{i}^i + (\mu(\nabla \otimes \underline{v}))_j \cdot \underline{i}^i] \cdot \underline{S}_j \end{aligned} \quad (3.24)$$

where \underline{i}^i is the base vector corresponding to the Cartesian coordinate x_i . This term is added to b_ϕ , where ϕ is the corresponding velocity vector component v_i .

3.6 DERIVATION OF PRESSURE EQUATION

Treatment of the coupling between pressure and velocities is an important aspect of the solution of the fluid flow equations (Raithby & Schneider [1979], Zedan & Schneider [1983], Latimer & Pollard [1985], Jang

et al. [1986]). Many solution algorithms have been developed to handle this coupling, and may be classified as either simultaneous or segregated.

The philosophy behind *simultaneous algorithms* (Caretto et al. [1972a], [1972b], Galpin et al. [1985], Vanka [1986]) is to solve for all variables simultaneously over the entire domain or over some subregion of the flow. This has the advantage that there is no need to explicitly derive a pressure equation. Such a procedure will produce the 'exact' solution of the linearised system of algebraic equations, but iteration is necessary to account for the non-linearity and explicit treatment of some terms, as explained in the previous section. However, the number of algebraic equations comprising the complete set of equations is often so large that the cost of a simultaneous solution becomes formidable.

In a *segregated approach* (Patankar & Spalding [1972], Patankar [1980], Van Doormaal & Raithby [1985], Issa [1986]) the algebraic representations of the equations for conservation of mass, momentum, energy, etc., are solved in sequence. In order to obtain the solution to the complete coupled set, each subset must be solved several times to account for the interequation coupling. This is not a large extra cost, since iteration is necessary anyway to allow for the non-linear nature of governing equations.

In the case of the compressible flow at non-negligible Mach number, the density may be regarded as one of the dependent variables and the pressure may be obtained from an equation of state. However, this approach is not useful for very low Mach number flows because the continuity equation can no longer be regarded as a density equation (in fact in the case of constant density this variable no longer features in the continuity equation) and some other means for obtaining the pressure field is needed. The way of deriving pressure from continuity and momentum equations is not so obvious. In what follows the generalised segregated approach and the particular example of the SIMPLE method of Patankar and Spalding [1972] for pressure calculation will be described, and the steps required to adapt them to the non-staggered arrangement of variables will be outlined (Peric [1985]). Firstly, the evaluation of the mass

fluxes, which prevents pressure-velocity decoupling when the colocated variable arrangement is used, will be described.

3.6.1 Evaluation of Mass Fluxes

In the colocated (non-staggered) variable arrangement, which is adopted in the present study, all the variables share the same grid position and control volume. This arrangement simplifies computer programming, minimises the amount of geometrical information required about the computational grid (Peric [1985]) and simplifies multigrid implementation (Smith [1990]). However, a special treatment is necessary to avoid a non-physical oscillatory pressure field, as will be described.

The discretised momentum and mass balance equations are written in the following form:

$$a_o \underline{v}_{p_0} - \sum_{j=1}^n a_j \underline{v}_{p_j} = b_{\underline{v}} - V_{p_0} \nabla p_{p_0} \quad (3.25)$$

$$\sum_{j=1}^n \rho \underline{v}_j \cdot \underline{S}_j = 0 \quad (3.26)$$

where the pressure term has been extracted from the source term (3.24) and Gauss's theorem has been applied, i.e.:

$$\sum_{j=1}^n p_j \underline{S}_j = V_{p_0} \nabla p_{p_0} \quad (3.27)$$

When a second-order $2h$ space-centred approximation of the pressure gradient on a colocated grid is used in expression (3.27), the pressure gradient cannot sense pressure oscillations which have a period $2h$ (Figure 3.6). The consequence of this anomaly can be appreciated by considering an inviscid flow in a straight channel in which velocity and pressure fields are uniform (Patankar [1980]). If in the numerical simulation the correct velocity field, but an incorrect 'checkerboard' pressure field with

variations of period $2h$ are specified as initial conditions (Figure 3.6), the pressure term in equation (3.25) would be zero everywhere, so the incorrect pressure field would be left unchanged.

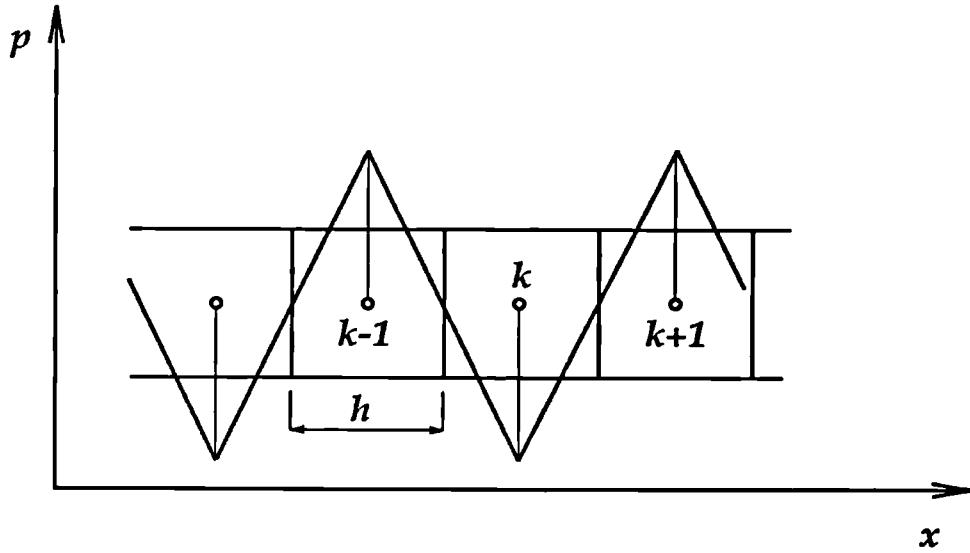


Figure 3.6 Checkerboard pressure field in straight channel

In earlier unsuccessful attempts to use the non-staggered variable arrangement, the velocity at the cell face \underline{v}_j in equation (3.26) was obtained by linearly interpolating the velocities at adjacent grid points P_o and P_j , thus:

$$\underline{v}_j = w_{P_o} \underline{v}_{P_o} + w_{P_j} \underline{v}_{P_j} = \overline{\underline{v}_{P_{oj}}} \quad (3.28)$$

As a consequence, the anomalous checkerboard pressure field persists, since the momentum and mass conservation equations are satisfied by such fields.

Hsu [1981], Prakash [1981] and Rhie [1981] suggested a different interpolation practice for cell-face velocities. A result of this practice is that the velocity used for the calculation of the mass flux depends 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 pressure coupling term such that:

$$\underline{v}_j = \overline{\underline{v}_{P_{Oj}}} + K_j \left\{ \frac{1}{2} [(\nabla p_{P_o})_{d_j} + (\nabla p_{P_j})_{d_j}] - \widetilde{\nabla p} \right\} \frac{\underline{S}_j}{|\underline{S}_j|} \quad (3.29)$$

where $(\nabla p)_{d_j}$ and $\widetilde{\nabla p}$ are defined as follows:

$$(\nabla p)_{d_j} = \nabla p \cdot \frac{\underline{d}_j}{|\underline{d}_j|} \quad (3.30)$$

$$\widetilde{\nabla p} = \frac{p_{P_j} - p_{P_o}}{|\underline{d}_j|} \quad (3.31)$$

The value for K_j is obtained from the momentum equation (3.25) (Rhie & Chow [1983]):

$$K_j = \frac{1}{2} \left[\left(\frac{V}{a_o} \right)_{P_o} + \left(\frac{V}{a_o} \right)_{P_j} \right] \quad (3.32)$$

Now a checkerboard pressure distribution in the inviscid flow example used earlier will result in non-zero values of the pressure term in (3.29), hence the mass fluxes calculated using these velocities will not satisfy the continuity constraint, even when the velocity field imposed at central nodes is correct. The mass imbalance will produce pressure corrections and will lead to the correct fields of both velocity and pressure.

The cell-face velocity is influenced by the pressure gradient $\widetilde{\nabla p}$ based on the $1h$ -centre difference through the pressure coupling term in equation (3.29), which can sense the pressure oscillations of the scale $2h$. This procedure ensures strong velocity-pressure coupling (Rhie & Chow [1983]).

Once a convergent solution is obtained, the pressure coupling term will vanish if the pressure gradient varies linearly in space, or it will become small in the limit of a fine grid, and the velocity at the cell face will be the same as the one calculated by (3.28), which implies that the value of the

multiplier K_j used is not important in such conditions. However, in general, the pressure gradient does not always vary linearly, nor is the mesh density always sufficient to resolve dramatic changes of the pressure field which may occur in some regions of the computational domain. A consequence is that the magnitude of the pressure term in such regions may be comparable to the value of the interpolated velocities, so that the cell-face velocity depends not only on the velocity field only, but also on the improperly-resolved pressure field. To avoid this, it is necessary either to refine the numerical grid in the regions where the magnitude of this term is significant, or to make the multiplier sufficiently small, so that the contribution of the pressure term to the cell face velocity, when the convergent solution is obtained, is negligible. Further discussion about this term will occur later, in Chapter 5.

In the first implementation of the method presented herein, the corrective velocity in expression (3.29) was taken to have the direction of vector \underline{d}_j . However, expression (3.29) is used to construct the pressure-correction equation (3.42), where the part of the corrective velocity based on the pressure difference across the face j is multiplied by the surface vector. When the angle between \underline{S}_j and \underline{d}_j is large, the coefficient a_j in the pressure correction equation is small, resulting in unnecessarily large corrections in that particular direction and leading to instabilities. Taking the corrective velocity to act in direction of \underline{S}_j , causes the pressure matrix to be better balanced, (the coefficients are more uniformly distributed), such that the pressure correction field is smoother and stable convergence is achieved.

3.6.2 Generalised Segregated Approach

Many problems of practical interest require the solution of the equations of motion for an incompressible fluid flow. Currently the most popular numerical methods solve the primitive-variable equations using the segregated approach, such as the SIMPLE algorithm of Patankar and Spalding [1972] and its variants. This section presents the generalised segregated approach of Van Doormaal & Raithby [1985] for solving the linear set of algebraic equations for velocity and pressure, which permits derivation of particular segregated methods.

In order to facilitate the presentation, equations (3.25) and (3.26), are assembled for all computational points and written in matrix notation, taking into consideration equation (3.29) for the colocated grid arrangement, as:

$$\mathbf{A}^v \mathbf{v} = \mathbf{b}^v - \mathbf{C}^p \mathbf{p} \quad (3.33)$$

$$\mathbf{M}^v \mathbf{v} + \mathbf{M}_{d,j}^p \mathbf{p} - \mathbf{M}^p \mathbf{p} = 0 \quad (3.34)$$

Given an estimate of pressure, denoted by p^* , the corresponding velocity v^* , which satisfies momentum equation, is given by:

$$\mathbf{v}^* = (\mathbf{A}^v)^{-1} \mathbf{b}^v - \mathbf{D}^p \mathbf{p}^* \quad (3.35)$$

where $\mathbf{D}^p = (\mathbf{A}^v)^{-1} \mathbf{C}^p$, which represents the influence of pressure on velocity. Since the pressure p^* is not in general correct, the mass conservation (3.34) will not be satisfied. It is important to notice that the velocity v^* is obtained through an iterative solution procedure, such that the inverse of \mathbf{A}^v is never calculated, since this exercise is memory and computer time expensive (Section 4.1). Therefore, expression (3.35) cannot be substituted directly into equation (3.34) to obtain the correct pressure. So in order to improve the estimate of the v^* velocity it is necessary to subtract an approximation of the effect of p^* on v^* , and add the approximate effect of an improved pressure estimate p^{**} , such that the improved estimate of velocity v^{**} , is given by:

$$v^{**} = v^* + \bar{\mathbf{D}}^p p^* - \bar{\mathbf{D}}^p p^{**} \quad (3.36a)$$

or

$$v^{**} = (\mathbf{K}^v)^{-1} (\mathbf{E}^v v^* + \mathbf{b}^v) - \bar{\mathbf{D}}^p p^{**} \quad (3.36b)$$

where \mathbf{A}^v is split into implicit, \mathbf{K}^v , and explicit, \mathbf{E}^v , parts, $\mathbf{A}^v = \mathbf{K}^v - \mathbf{E}^v$ and $\bar{\mathbf{D}}^p$ is an approximation of \mathbf{D}^p , $\bar{\mathbf{D}}^p = (\mathbf{K}^v)^{-1} \mathbf{C}^p$.



By requiring that the v^{**} velocities satisfy mass conservation, given by equation (3.34), the following equation for p^{**} results:

$$A^p p^{**} = b^p \quad (3.37)$$

where:

$$A^p = (M^v \bar{D}^p + M_{d_j}^p - M^p) \quad (3.38)$$

$$b^p = -M^v (v^* + \bar{D}^p p^*) \quad (3.39)$$

Solving equation (3.37) for p^{**} the solution for v^{**} is readily determined from equation (3.36). It is important to note that if \bar{D}^p is chosen to be identical to D^p , then p^{**} and v^{**} will satisfy equations (3.33) and (3.34) exactly. However, because an approximate evaluation of D^p is used to obtain p^{**} and v^{**} these solutions will not satisfy equations (3.33) and (3.34) unless the choice for p^* happens to be correct. Since this is generally not true, the resulting segregated approach is implemented by performing the following sequence of steps:

- (1) Guess a pressure field p^*
- (2) Evaluate the coefficients of the momentum equations (3.33) and solve for v^* using p^* .
- (3) Evaluate the coefficients of equation (3.37)
- (4) Obtain the source term of equation (3.37) and solve for p^{**} .
- (5) Calculate the improved estimate for velocity v^{**} from equation (3.36)
- (6) Using the p^{**} as the new p^* , return to step 2. Repeat this cycle until convergence is achieved.

In the approach described above the solution of the tentative velocity v^* , and approximate pressure p^{**} , are determined separately, in an uncoupled manner. Without prescribing how to split matrix A^v , the method can be viewed as a generalisation of the segregated approach for solving incompressible fluid flows.

3.6.3 The SIMPLE Algorithm

Using the segregated approach described in the previous section, the SIMPLE method of Patankar and Spalding [1972] can be introduced as follows. The implicit matrix K^v is taken to be the diagonal of A^v . All equations are rewritten in terms of pressure correction $p' = p^{**} - p^*$ so that equation (3.35) is subtracted from (3.36) to give:

$$v^{**} = v^* - \bar{D}^p p' \quad (3.40)$$

and equation (3.37) becomes:

$$A^p p' = b^p \quad (3.41)$$

The contribution of the term $(M^v \bar{D}^p + M_{d_j}^p)$ in equation (3.38) to the coefficients of the matrix A^p can be negative, and the system (3.41) is thus no longer diagonally dominant. Moreover, its implicit treatment will extend the computational molecule, increasing memory demands. However, since the influence of this term is small if the grid is not severely non-orthogonal, so it is treated explicitly in the present study. Now the coefficients of matrix A^p and source b^p of pressure correction equation are as follows:

$$\begin{aligned} a_j^p &= -\rho \left(\frac{\bar{V}_p}{a_o^v} \right) \frac{|S_j|}{|d_j|} \\ a_o^p &= - \sum_{j=1}^n a_j^p \\ b^p &= \sum_{j=1}^n F_j^* \end{aligned} \quad (3.42)$$

where F_j^* is cell face mass flux defined by equations (3.4) and (3.32) with respect to velocity v^* and pressure p^* . When the velocity is corrected according to (3.40), the flux is updated as follows:

$$F_j^{**} = F_j^* - \alpha_p^p (p_{Pj} - p_{P\delta}) \quad (3.43)$$

and this new flux satisfies the mass conservation equation (3.26), while the one based on equation (3.4) and new velocity and pressure field in general does not.

The approximations that are introduced in the segregated approach described above often result in an overestimation of the magnitude of p' which in turn leads to slow convergence or divergence of the method. To remedy this, the correction of pressure is under-relaxed by:

$$p^{**} = p^* + \alpha_p p' \quad (3.44)$$

where recommendations for α_p can be found in Peric [1985], for example.

3.7 BOUNDARY CONDITIONS AND THEIR IMPLEMENTATION

The boundary conditions used in the present study and the treatment necessary to incorporate them in the discretised equations are presented in this section.

3.7.1 Inlet Boundaries Definition

The inlet boundaries are of Dirichlet type and the values of all dependent variables at this boundary are known. Implementation is simple and straightforward. The inlet boundary values at the cell faces that coincide with the boundary are already available, without any interpolation, and they are used in calculation of fluxes and gradients at P_o (Figure 3.7).

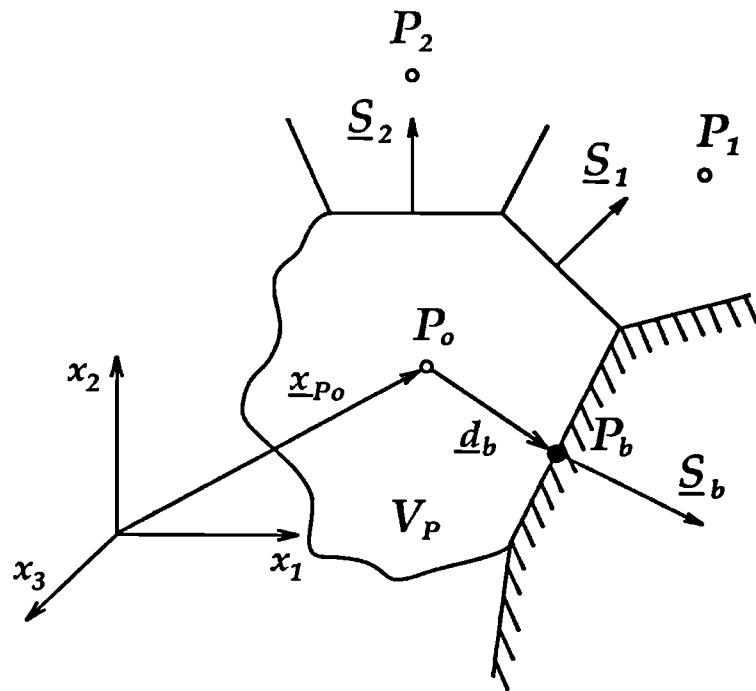


Figure 3.7 Control volume next to the boundary

3.7.2 Outlet Boundaries Definition

An outlet boundary should be placed far downstream from the region of interest, at a location where the flow is everywhere directed outwards so that any inaccuracy in estimating the outlet conditions will not propagate far upstream, provided that the Reynolds number is large. It is often adequate to set boundary values equal to the values at the immediate upstream neighbour e.g. $\phi_{P_b} = \phi_{P_o}$ for the situation shown in Figure 3.7 which is equivalent to the zero gradient boundary condition in the direction of d_b . Otherwise linear or quadratic extrapolation can be used. Such defined outlet boundaries are boundary conditions of Neumann type.

3.7.3 Symmetry Boundaries Definition

Numerical implementation of symmetry boundary conditions can be interpreted (Demirdzic [1991]) as follows.

The considered flow domain has its mirror image across the symmetry boundary. The symmetry boundary conditions should therefore be implemented in such a way that fields of dependent variables are identical to ones obtained if the whole domain (i.e. the actual one and its mirror image) is meshed and calculated (Figure 3.8). Now the numerical implementation of symmetry boundary conditions becomes simple as follows. The geometrical properties and dependent variable values stored at boundary computational point P_b are substituted with one of the mirror image of P_o , i.e.:

$$\begin{aligned} \underline{x}_{Pb} &= \underline{x}_{Pm} = \underline{x}_{Po} + 2 (\underline{d}_b \cdot \underline{S}_b) \frac{\underline{S}_b}{|\underline{S}_b|^2} \\ \underline{v}_{Pb} &= \underline{v}_{Pm} = \underline{v}_{Po} - 2 (\underline{v}_{Po} \cdot \underline{S}_b) \frac{\underline{S}_b}{|\underline{S}_b|^2} \\ \phi_{Pb} &= \phi_{Po} \end{aligned} \quad (3.45)$$

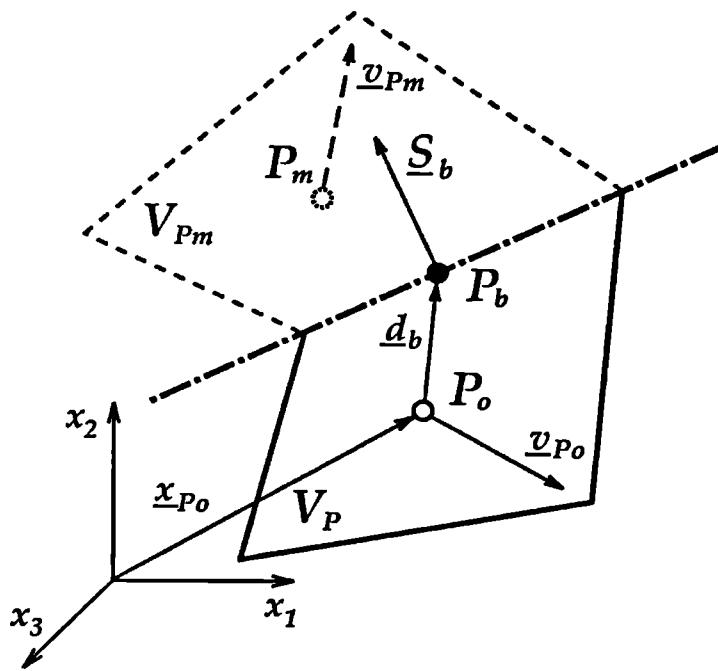


Figure 3.8 Symmetry plane boundary

Having in mind that now all relevant data of the mirror image cell P_m is stored at the boundary node P_b , the cell face and the cell next to

the symmetry boundary will be treated as an ordinary cell as described in Section (3.3).

3.7.4 Impermeable No-Slip Walls

The fluid velocity at a rigid impermeable wall is the same as the velocity of the wall (no-slip boundary conditions). These conditions are easy to impose. However, in the case of turbulent flows the calculation of the stresses on the wall needs special treatment (Schlichting [1968], Launder & Spalding [1974], Gosman & Ideriah [1976]), since the standard $k-\varepsilon$ model of turbulence becomes inadequate in near wall regions. Wall functions formulae in conjunction with the $k-\varepsilon$ turbulence model (Launder & Spalding [1974]) are used in the present study to bridge this region.

The fluxes in the momentum equation represent forces that act on the control volume in the direction of velocity components. The resultant shear surface force acting on the cell face next to the impermeable wall is expressed as a function of the gradient of the velocity component parallel to the wall in the direction normal to the wall:

$$T_w = -\mu_w \frac{(\underline{v}_{p_o})_p - \underline{v}_w}{\delta n} |\underline{S}_b| \quad (3.46)$$

where $(\underline{v}_{p_o})_p = \underline{v}_{p_o} - (\underline{v}_{p_o} \cdot \underline{S}_b) \frac{\underline{S}_b}{|\underline{S}_b|^2}$ is the velocity vector component parallel to the wall. Similarly, the integral of wall heat flux in the energy equation (2.14) is given by:

$$Q_w = \lambda_w \frac{T_{p_o} - T_w}{\delta n} |\underline{S}_b| \quad (3.47)$$

In the above expressions δn is the normal distance from the wall:

$$\delta n = (\underline{d}_b \cdot \underline{S}_b) \frac{1}{|\underline{S}_b|} \quad (3.48)$$

v_w is the wall velocity, T_w wall temperature and the coefficients μ_w and λ_w are determined from a two-part 'universal' velocity profile expression (Launder & Spalding [1974]):

i) viscous sublayer ($y^+ \leq 11.6$):

$$\begin{aligned}\mu_w &= \mu \\ \lambda_w &= \lambda\end{aligned}\quad (3.49)$$

ii) fully turbulent layer ($y^+ > 11.6$):

$$\begin{aligned}\mu_w &= \frac{\rho C_{\mu}^{1/4} k_{P_o}^{1/2} \delta n}{\frac{1}{\kappa} \ln(Ey^+)} \\ \lambda_w &= \frac{\mu c_p}{\sigma_{h,t}} \frac{y^+}{\frac{1}{\kappa} \ln(Ey^+) + P\left(\frac{\sigma_h}{\sigma_{h,t}}\right)}\end{aligned}\quad (3.50)$$

where y^+ is the dimensionless distance of the node P_o from the wall, defined as:

$$y^+ = \frac{\rho C_{\mu}^{1/4} k_{P_o}^{1/2} \delta n}{\mu} \quad (3.51)$$

The $P(\sigma_h, \sigma_{h,t})$ function is the viscous sublayer resistance factor expressing the contribution of the viscous sublayer to the total thermal resistance and is given for smooth walls by Jayatillaka [1969]:

$$P\left(\frac{\sigma_h}{\sigma_{h,t}}\right) = 0.9 \left(\frac{\sigma_h}{\sigma_{h,t}} - 1\right) \left(\frac{\sigma_h}{\sigma_{h,t}}\right)^{-1/4} \quad (3.52)$$

In the above equations $\kappa = 0.4187$ is von Karman's constant, c_p is specific heat at constant pressure, $\sigma_h (=0.7)$ and $\sigma_{h,t} (=0.9)$ are the laminar and turbulent Prandtl numbers respectively, and E is an integration constant, which depends on the wall roughness. For smooth impermeable walls E is assigned a constant value of 9.0 (Launder & Spalding [1974]).

Thus, for the v_i momentum and energy equations the shear stress and heat flux integrals over boundary cell faces, appearing in equations such as (3.24) and (3.8), are replaced by the T_{w_i} and Q_w defined by (3.46) and (3.47).

The equations for k and ε also need special treatment for the near wall cells. In the equation for k , the diffusion flux through the wall cell face D_j is taken as zero (Launder & Spalding [1974]), and the rate of generation of turbulent kinetic energy G , obtained from the velocity log-law expression at the near wall, is given by:

$$G = |\underline{\tau}_w| \frac{C_\mu^{1/4} k_{P0}^{1/2}}{\kappa \delta n} \quad (3.53)$$

where $\underline{\tau}_w$ is the shear force per unit area:

$$\underline{\tau}_w = \frac{T_w}{|S_b|} \quad (3.54)$$

The dissipation rate ε at the near-wall nodes is fixed to the value:

$$\varepsilon = \frac{C_\mu^{3/4} k_{P0}^{3/2}}{\kappa \delta n} \quad (3.55)$$

which replaces the solution of the ε equation there.

3.8 SOLUTION PROCEDURE FOR THE COUPLED EQUATION SETS

It is now possible to define the sequence of operations which, when repeated as many times as necessary, leads to the solution of the discretised governing equations. It can be summarised as follows:

- (0) Set all the field values by an initial guess.
- (1) Assemble coefficients and sources of the momentum equation (3.33) using the current available estimates of dependent variables.

- (2) Solve the linearised momentum equations to obtain \mathbf{v}^* .
- (3) Solve the pressure-correction equation (3.41) to obtain p' .
- (4) Correct the velocity using expression (3.40), the mass fluxes using equation (3.43) and the pressure using equation (3.44) to obtain the \mathbf{v}^{**} , F_j^{**} and p^{**} .
- (5) If the flow is turbulent, solve the equations for k and ε , and calculate the new μ_{eff} .
- (6) Solve the linearised energy equation and use the equations of state to obtain the new temperature and density fields.
- (7) If the convergence criterion is not satisfied return to step 1.

In order to promote stability of the solution method, under-relaxation is often necessary. This may be introduced in an implicit (so called 'inertial') manner, by re-writing equation (3.2) as:

$$\frac{a_o}{\alpha_\phi} \phi_{P_o}^k - \sum_{j=1}^n a_j \phi_{P_j}^k = b_\phi + (1-\alpha_\phi) \frac{a_o}{\alpha_\phi} \phi_{P_o}^{k-1} \quad (3.56)$$

where α_ϕ is an under-relaxation factor for variable ϕ , with a value between 0 and 1, and k is the outer iteration counter. In addition, this enhances the diagonal dominance of the linearised equations, which improves the rate of convergence of iterative linear equation solvers (Section 4.2.1).

The convergence of outer the iterations can be defined in various ways. In the present study the normalised sum of absolute residuals R_ϕ^k (see Section 4.2) at all computational nodes is used as the test:

$$R_\phi^k = \frac{\sum_{i=1}^N |r_\phi^k|}{M_\phi} \quad (3.57)$$

where residual r_ϕ^k at node P_o and iteration k is calculated as the difference between the right and the left hand side of equation (3.56). The normalising factor M_ϕ is obtained by summing the relevant inflow fluxes of the conserved variable Φ over all boundaries (Peric [1985], STAR-CD Manual [1991]).

3.9 CLOSURE

The finite volume discretisation technique has been adopted in the present study. It has been decided to use unstructured numerical grids comprised of control volumes of arbitrary topology, which offer great flexibility and hence good quality meshing for complicated geometrical configurations. The most important feature of this space discretisation, relevant for the present study, is that it enables efficient local grid refinement only in the regions where it is necessary, without disturbing the rest of the computational mesh.

A novel discretisation of the model transport equation has been carried out in physical space, in a way which is easy to understand and implement, and which can cope with an arbitrary cell topology. The final form of the discretised transport equations is a set of algebraic equations, one for each computational node, in which a number of coefficients multiplying values of the dependent variable at neighbouring nodes appear. The number of neighbouring nodes involved depends on the cell topology and the spatial interpolation practices used to express the spatial variations of the dependent variables in the neighbourhood of the cell considered. In this study a linear spatial variation has been assumed and determined using the least-squares method, involving all nearest neighbours. However, in the evaluation of the convection fluxes in the presence of highly non-linear spatial variation of the dependent variables, modifications are required in order to avoid non-physical oscillations in the numerical solution. A flux blending method, combining good features of first-order UDS and second-order CDS, has been implemented in a deferred-correction manner, as suggested by Khosla & Rubin [1974]. The colocated arrangement, in which all dependent variables are stored at the centres of the control volumes, has been adopted since it is the simplest arrangement, and perhaps the most economical in general situations from the numerical solution point of view.

An interpolation practice for evaluating of cell face velocity and mass fluxes, first suggested by Rhie [1981], enables avoidance of pressure-velocity decoupling problems when the colocated arrangement and second-order

space-centred pressure gradient calculation are employed. The SIMPLE algorithm of Patankar and Spalding [1972] has been chosen for pressure-velocity coupling and presented in light of the generalised segregated approach of Van Doormaal and Raithby [1985] which has been introduced in the context of the discretisation practices presented in this study.

The most frequently-encountered boundary conditions and the treatment necessary to incorporate them into the discretised equations have been described. Standard wall functions have been employed in the treatment of the turbulent near-wall flow. Finally, the sequence of operations of a typical iteration cycle has been presented.

In the next chapter some basic iterative techniques for solving the system of linear equations will be discussed, with particular emphasis on the multigrid method, the most efficient and general iterative technique known today.

Chapter 4

Multigrid Methodology

4.1 INTRODUCTION

In the previous chapter, it was shown that, by the process of the finite volume discretisation, the problem of solving the governing equations of Chapter 2 may be reduced to one of solving a system of *coupled nonlinear algebraic equations*. In order to facilitate the use of well-established methods for solution of a system of *linear algebraic equations*, the aforementioned system has to be *linearised*, and set of equations for each dependent variable temporarily decoupled (Section 3.8) by assuming that coefficients and source terms are known (calculated using dependent variable values from the previous outer iteration). Because of the nature of the described discretisation practice (Section 3.5), the coefficient matrix of the linearised system is sparse (i.e. there are relatively few nonzero elements).

Existing methods for solving systems of linear algebraic equations fall into two main categories: *direct* and *iterative* methods. Direct methods lead to the 'exact' solution of the linear algebraic system in a finite number of arithmetic operations. Iterative methods begin with an initial guess of the solution and then proceed to improve the current approximation by a succession of simple updating steps called '*inner iterations*' or *relaxation*, leading to the 'exact' solution after an infinite number of steps. In practice, the number of arithmetic operations of a direct method can be very high, leading to a solution which is strongly dependent on the finite arithmetic of the computer. The total number of operations of an iterative method is

limited by a finite level of the convergence, again related to the finite number of digits of the computer, and is often much smaller than the number of arithmetic operations of direct methods. When the number of algebraic equations to be solved is very large, only iterative methods are acceptable. In addition, since the system of linearised algebraic equations is only an approximation of the original system of nonlinear equations, there is no point in solving it to a very tight tolerance, and a sufficiently improved solution can sometimes be achieved in just a few inner iterations. This cannot be exploited by direct methods.

A large number of linear iterative methods are available with different levels of complexity and rates of convergence. Most of them suffer from two important limitations, which are of the following kind:

- The convergence rate slows dramatically after the first few iterations, making achievement of a converged solution very time consuming and therefore expensive.
- The total computational work is proportional to N^m , where N is the number of algebraic equations (number of control volumes comprising the computational domain), and m is greater than one. This is a particularly severe problem, since the realisation of an accurate solution, which usually requires very fine grids, can be at best extremely expensive and at worst not feasible.

The problem of converging the system of linearised algebraic equations is not the only one facing someone who tries to solve the system of nonlinear algebraic equations. The final solution will be obtained by converging the outer iterations. The limitations listed above, are characteristic for iterative algorithms designed to handle the nonlinear nature of the governing equation and interequation coupling as well (Sections 3.6.3 and 3.8). In order to promote convergence of these methods, it is often necessary to under-relax discretised equations, e.g. in the manner described in Section 3.8. Their convergence rate depends, among other things, on the nature of the particular fluid flow, on the interequation coupling, the choice of linear iterative method for the solution of linearised systems, the choice of under-relaxation parameters, and also on

the initial guess. A general, formal mathematical analysis of the convergence properties of these methods does not exist, and numerical experiment is the only tool which gives insight into the behaviour of these methods.

Multigrid (MG) methods have evolved from attempts to correct the aforementioned limitations. These methods are applicable to both linear and nonlinear systems and indeed are the most efficient and general iterative techniques known today for solving problems whose nature is elliptic. Their convergence rate is, in the ideal case, independent of the number of computational points. In combination with successive grid refinement, or nesting, they can provide solutions with an iteration error smaller than the discretisation error in one to three multigrid cycles (Brandt [1982], Hackbusch [1985]). Non-elliptic problems can also be handled, if sufficient numerical ellipticity is present (Mulder [1989]). This means that, on the scale of the grid spacing, there is a reasonable amount of coupling between neighbouring points or cells. This coupling is then exploited to remove oscillatory components of the error by a suitable smoother. The multigrid method fails if this coupling becomes too small in one direction. The basic idea of multigrid methods is derived from the fact that the algebraic system to be solved does not stand by itself, but is actually an approximation of continuous equations, and therefore can itself be similarly (temporarily) approximated by another, much simpler algebraic system for the purpose of accelerating the solution. In practice this means that it is possible to build a hierarchy of numerical meshes representing the same computational domain, each of them progressively having higher numerical resolution, ranging from very coarse to very fine. By establishing a dialogue among these computational meshes, it is possible to develop an extremely efficient algorithm, where coarse grids provide fast propagation of information from boundaries to the interior and the fine grids supply the desired accuracy by resolving local detailed features of the fluid flow. The whole concept is quite general and has a wide range of applications (McCormick [1987]). The method can be used not only to accelerate linear iterative methods but also, for example, to solve nonlinear problems or to devise adaptive discretisation algorithms: both of these facets will be demonstrated later.

As pointed out in Sections 1.1 and 1.2, the method of integration of MG into an iterative scheme designed to handle the nonlinear governing equations and interequation coupling is crucial for its convergence performance. The details of the implementation are closely associated with the fundamental solution algorithm (e.g. SIMPLE algorithm of Patankar & Spalding [1972]), used to couple the various sets of algebraic equations, each pertaining to a related flow variable. Imbedding the SIMPLE algorithm as a smoother in the MG method by Sivaloganathan & Shaw [1988] and Peric et al. [1989] has proved to be superior to the implementation of Phillips et al. [1985], which used MG to accelerate the convergence of the linear solver for each set of linearised algebraic equations associated with a relevant flow variable. Thus, the approach of Sivaloganathan & Shaw [1988] was adopted in the present study.

In the next section, iterative solvers are briefly reviewed, and the convergence of the Gauss-Seidel technique is analysed for a simple model problem. This analysis exposes the sources of the problems described above, and points at the multigrid method for overcoming them, which is described in Section 4.3. In Section 4.4 the coarse grid equation required by multigrid solvers is derived in the framework of a *Full Approximation Storage* (FAS) scheme. Boundary conditions for the coarse grid are then discussed in Section 4.5. Finally, the overall multigrid algorithm is described and its major components examined in Section 4.6.

4.2 ITERATIVE SOLUTION TECHNIQUES FOR SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS

4.2.1 Basic Iterative Methods

In matrix notation, a system of N linear algebraic equations is written as:

$$\boxed{A \phi = b} \quad (4.1)$$

where A is the real $N \times N$ nonsingular coefficient matrix and ϕ and b are the exact solution and the source column vector, respectively. Many schemes for the iterative solution of the set of linear equations are based on splitting of the matrix A into implicit, K , and explicit, E , components (Varga [1962], Meijerink & Van der Vorst [1977], Hageman & Young [1981]) thus:

$$A = K - E \quad (4.2)$$

Substitution of equation (4.2) into (4.1) yields the following equation for the solution vector $\phi^{(k+1)}$ at the new iteration level ($k+1$) in terms of the values at the old iteration level (k):

$$K \phi^{(k+1)} = E \phi^{(k)} + b \quad (4.3)$$

or

$$\phi^{(k+1)} = M \phi^{(k)} + h \quad (4.4)$$

where $M = K^{-1}E$ is the real $N \times N$ *iteration matrix* (Jennings [1977]) and $h = K^{-1}b$ is the associated known vector. The expression (4.4) stands for a *linear stationary iterative method of the first degree* for the solution of (4.1) which is said to be convergent if the solution vector $\phi^{(k+1)}$ tends (converges) to the exact solution ϕ as k tends to infinity. The method is of *first degree* since $\phi^{(k+1)}$ depends explicitly only on $\phi^{(k)}$. The method is *linear* since neither M nor h depend on $\phi^{(k)}$, and it is *stationary* since neither M nor h depend on k .

There are two important measures of $\phi^{(k)}$ as an approximation to ϕ . One is the *error* (or *algebraic error*) column vector, which is given by:

$$\epsilon^{(k)} = \phi^{(k)} - \phi \quad (4.5)$$

Its size can be measured by any of the standard vector norms. The most commonly-used norms are of the form:

$$\|\varepsilon\|_p = \left(\sum_{j=1}^N |\varepsilon_j|^p \right)^{1/p} \quad (4.6)$$

The norm (4.6) for $p=2$ is known as Euclidean norm. Unfortunately, this error is as inaccessible as the exact solution. A computable measure of how well $\phi^{(k)}$ approximates ϕ is the *residual* $r^{(k)}$, given by:

$$r^{(k)} = b - A \phi^{(k)} \quad (4.7)$$

Combining (4.1) and (4.7) it appears that the residual $r^{(k)}$ acts as a source of the algebraic error $\varepsilon^{(k)}$, i.e.

$$A\varepsilon^{(k)} = r^{(k)} \quad (4.8)$$

As the exact solution is approached, $r^{(k)}$ tends to the zero vector, so that monitoring $r^{(k)}$ indicates the progress towards convergence. Note that by the uniqueness of the solution ϕ , $r^{(k)}=0$ if and only if $\varepsilon^{(k)}=0$.

Combining (4.7) and (4.3), the iterative method (4.4) can be written as:

$$\phi^{(k+1)} = \phi^{(k)} + K^{-1}r^{(k)} \quad (4.9)$$

From the definition of an iterative method, such as (4.4), the error vector $\varepsilon^{(k)}$ can be expressed as:

$$\varepsilon^{(k)} = M\varepsilon^{(k-1)} = M^k \varepsilon^{(0)} \quad (4.10)$$

The error vectors $\varepsilon^{(k)}$ tend to the zero vector for all $\varepsilon^{(0)}$ if and only if the *spectral radius* (being the eigenvalue largest in magnitude) $\rho(M)$ of the matrix M is less than unity. This is the necessary and sufficient condition for convergence of the method defined by (4.4). For many iterative schemes a rule of thumb is that convergence may be expected if A is an M-matrix, i.e. if the coefficients of the matrix A fulfil the following criterion:

$$a_{ii} \geq - \sum_{j=1}^N a_{ij}, a_{ij} \leq 0, j \neq i \quad (4.11)$$

or if the matrix A satisfies the Scarborough criterion (Scarborough [1958]):

$$a_{ii} \begin{cases} \geq \sum_{j=1}^N |a_{ij}| & \text{at all locations} \\ & j \neq i \\ > \sum_{j=1}^N |a_{ij}| & \text{at least at one location} \end{cases} \quad (4.12)$$

Matrices exhibiting property defined by (4.12) are called *diagonally dominant matrices*.

The term $\rho(M)$ is also called the *convergence factor* when it appears in the context of iterative methods, and it has some useful interpretations. It is roughly the worst factor by which the error is reduced by each relaxation sweep. The *asymptotic convergence rate* CR_{∞} , defined as the inverse of the number of iterations to reduce the error by an order of magnitude in the limit as k tends to ∞ depends on the spectral radius, according to:

$$CR_{\infty} = -\log_{10}(\rho(M)) \quad (4.13)$$

The properties of the iteration matrix M depend on the split of A into K and E . Generally, the more K resembles A , the faster the method will converge (Meijerink & Van der Vorst [1977]). However, K must also be chosen so that equation (4.4) is easy to solved at each iteration, without excessive extra work or storage requirements.

Choices of K and E corresponding to some well established iterative solvers are listed in Table 4.1, where ω is a relaxation factor, D is the diagonal matrix and L and U are the strictly lower and upper triangular matrices, such that:

$$A = D - L - U \quad (4.14)$$

The spectral radii of the iteration matrices given in Table 4.1 are taken from the book of Ames [1979]. These were presented for a model Dirichlet problem for the Laplace equation applied on the square $\pi \times \pi$ with the mesh spacing π/n . The resulting algebraic system was obtained using five point finite difference discretisation. As can be seen from Table 4.1, the spectral radii approach 1 as the number of computational points n in each direction increases, so convergence worsens as the mesh becomes finer. The reasons for this behaviour will be explained in Section 4.2.4. The diagonal dominance defined by (4.12) is sufficient, but not necessary, for convergence of the Jacobi and Gauss-Seidel method (Scarborough [1958]).

ITERATIVE SCHEME	K	E	$\rho(M)$
Jacobi	D	$L+U$	$\cos(\pi/n)$
Gauss-Seidel	$D-L$	U	$\cos^2(\pi/n)$
Successive overrelaxation - SOR	$\frac{1}{\omega}D-L$	$U + \frac{(1-\omega)}{\omega}D$	$\frac{1-\sin(\pi/n)}{1+\sin(\pi/n)}$

Table 4.1 Properties of some well established iterative methods (Ames [1979])

An approximation of the coefficient matrix by the sum of a lower and upper triangular and diagonal matrix, as in the Table 4.1 methods, is not the only way of generating an iterative scheme. In the following sections two other iterative techniques will be presented. One is based on an incomplete LU factorisation, and the other on so called 'polynomial acceleration'.

4.2.2 Incomplete LU Factorisation

A large group of iterative schemes is based on approximate LU factorisation. The matrix A can be written as the product of a lower L and an upper U triangular matrices:

$$A=LU \quad (4.15)$$

Note that L and U now have a different meaning from in equation (4.14). The elements l_{ij} and u_{ij} of L and U can be calculated recursively column-by-column as follows:

$$\left. \begin{aligned} l_{ii} &= \frac{1}{u_{ii}}(a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}) \\ u_{ij} &= \frac{1}{l_{ii}}(a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}) \\ l_{ji} &= \frac{1}{u_{ii}}(a_{ji} - \sum_{k=1}^{i-1} l_{jk}u_{ki}) \end{aligned} \right\} \begin{array}{l} j=i+1, \dots, N \\ i=1, \dots, N \end{array} \quad (4.16)$$

The decomposition (4.16) is not unique. However, if the values of the diagonal elements of matrix U (or L) are specified and different from zero, the decomposition will be unique.

Unfortunately, for large sparse matrices, the calculation of L and U is very consuming of both time and storage since many of the elements of A which are zero become nonzero in L and U , and so the desirable sparseness of the original matrix is lost.

In Section 4.2.1, we assumed that any basic method can be defined uniquely in terms of a splitting matrix K . In this section we consider those methods for which the matrix K has the form $K=LU$. An approximate factorisation of matrix A is to pick L and U to be lower and upper triangular matrices, respectively, but such that the product LU only approximates A . Usually this is done by defining a particular sparsity pattern for L and U and then determining the nonzero elements of L and U so that the product LU approximates A as closely as possible. This approach encompasses a family of iterative techniques, called incomplete LU factorisation, that differ mainly in the choice for the matrices L and U . Some members of this family can be found in Dupon et al. [1968], Stone [1968] and Meijerink and Van der Vorst [1977, 1981].

The incomplete Cholesky factorisation used in the present study follows that for symmetric and positive definite matrices described by

Kershaw [1978]. The decomposition is based on equation (4.16), where u_{ii} is chosen to be the same as l_{ii} , and the sparsity pattern, which is to be imposed on L and U , is the same as one of matrix A . A simple modification of (4.15) in which square roots are avoided is as follows:

$$A=LDU \quad (4.17)$$

where the off diagonal entries of D are zero and L , D and U are determined recursively as follows:

$$\left. \begin{aligned} l_{ji} &= a_{ji} - \sum_{k=1}^{i-1} l_{jk} d_{kk} u_{ki} \\ u_{ij} &= a_{ij} - \sum_{k=1}^{i-1} l_{ik} d_{kk} u_{kj} \\ d_{ii} &= (l_{ii})^{-1} = (u_{ii})^{-1} \end{aligned} \right\} \begin{array}{c} j=i, \dots, N \\ i=1, \dots, N \end{array} \quad (4.18)$$

As one proceeds with this algorithm, whenever ij corresponds to a location where $a_{ij}=0$, l_{ij} or u_{ij} is set to zero. In practice, these elements of L and U are neither calculated nor stored.

4.2.3 Polynomial Acceleration Techniques

The polynomial procedure is one of many approaches that can be used to accelerate the convergence of basic iterative methods (Hageman & Young [1981]). Here, the key features of conjugate gradient (CG) acceleration will be described.

For a Symmetric and Positive Definite (SPD) matrix A (that is, $(\phi, A\phi) > 0$, for any choice of ϕ) the CG method is an iterative procedure that converges to the exact solution in at most N iterations in the absence of rounding errors. If the residual $r^{(k)}$ is expanded as a series based on the eigenvectors of A , it is found that each step of the conjugate gradient method has the effect of approximately eliminating the contribution from one eigenvector. Consequently, if the matrix A has only m distinct

eigenvalues, where m can be much smaller than N , the conjugate gradient method obtains the exact answer after only m iterations.

The efficiency of the original conjugate gradient method (Hestenes & Stiefel [1952]) is not, however, very impressive compared to the performance of the basic iterative techniques described in Section 4.2.1. It derives its popularity from the possibility of convergence acceleration by preconditioning (Meijerink & Van der Vorst [1977, 1981], Kershaw [1978]). The idea behind preconditioning is to premultiply equation (4.1) so that the eigenvalues of the resulting matrix are bunched more closely together than those of A and the spread of the eigenvalues ($\rho(A)_{\min}, \rho(A)_{\max}$) is thereby reduced. As a consequence, the subsequent application of the conjugate gradient method to such a 'preconditioned' system converges faster. The best choice for the preconditioner would be A^{-1} , since then the solution follows trivially. A good choice is K^{-1} , which appears in (4.3).

The method for symmetric matrices (e.g. pressure equation (3.41)) used in the present study is described in detail by Kershaw [1978], and only an outline will be given here. The preconditioning matrix is the incomplete Cholesky decomposition of A :

$$K = LDL^T \quad (4.19)$$

where L and D are defined by equation (4.18), taking care that $u_{ij}=l_{ji}$ and forcing L to have the same sparsity pattern as A . The resulting algorithm is:

$$\left. \begin{array}{l} r^{(0)} = b - A \phi^{(0)} ; p^{(0)} = K^{-1} r^{(0)} ; \beta^{(0)} = 0 \\ \beta^{(k)} = \frac{(r^{(k)}, K^{-1} r^{(k)})}{(r^{(k-1)}, K^{-1} r^{(k-1)})} \\ p^{(k)} = K^{-1} r^{(k)} + \beta^{(k)} p^{(k-1)} \\ \alpha^{(k)} = \frac{(r^{(k)}, K^{-1} r^{(k)})}{(p^{(k)}, A p^{(k)})} \\ \phi^{(k+1)} = \phi^{(k)} + \alpha^{(k)} p^{(k)} \\ r^{(k+1)} = r^{(k)} - \alpha^{(k)} A p^{(k)} \end{array} \right\} k = 0, 1, 2, \dots \quad (4.20)$$

where k is the counter of inner iterations, and the scalar product of two vectors x and y is given symbolically by (x,y) .

For many practical applications the restriction of CG to SPD systems is a severe drawback. The most obvious way to apply the CG method on nonsymmetric systems is to symmetrise a general matrix equation (4.1) by replacing it with (Kershaw [1978]):

$$A^T A \phi = A^T b \quad (4.21)$$

This approach is not, however, generally recommended, because if A is ill-conditioned, $A^T A$ will be even more so. However, Khosla and Rubin [1981] reported good performance by symmetrising the system of discretised Navier-Stokes equations in stream function-vorticity form governing incompressible viscous flow past a circular cylinder at $Re=100$ as in (4.21) and applying the SIP (Stone [1968]) -preconditioned CG method. Today, there is a large group of CG based solvers which are designed to be applied to nonsymmetric systems. The bi-conjugate gradient method of Fletcher [1976], ORTHOMIN method of Vinsome [1976], CGS method of Sonneveld et al. [1985] and CGSTAB of Van der Vorst and Sonneveld [1990] are some of the most often used and discussed in literature. The preconditioned bi-conjugate gradient method used in the present study operates as follows:

$$\left. \begin{aligned} r^{(0)} &= b - A \phi^{(0)}; r_t^{(0)} = b - A^T \phi^{(0)}; p^{(-1)} = p_t^{(-1)} = 0; \\ \beta^{(k)} &= \frac{(r_t^{(k)}, K^{-1} r^{(k)})}{(r_t^{(k-1)}, K^{-1} r^{(k-1)})} \\ p^{(k)} &= K^{-1} r^{(k)} + \beta^{(k)} p^{(k-1)} \\ p_t^{(k)} &= (K^T)^{-1} r_t^{(k)} + \beta^{(k)} p_t^{(k-1)} \\ \alpha^{(k)} &= \frac{(r_t^{(k)}, K^{-1} r^{(k)})}{(p_t^{(k)}, A p^{(k)})} \\ \phi^{(k+1)} &= \phi^{(k)} + \alpha^{(k)} p^{(k)} \\ r^{(k+1)} &= r^{(k)} - \alpha^{(k)} A p^{(k)} \\ r_t^{(k+1)} &= r_t^{(k)} - \alpha^{(k)} A^T p_t^{(k)} \end{aligned} \right\} k = 0, 1, 2, \dots \quad (4.22)$$

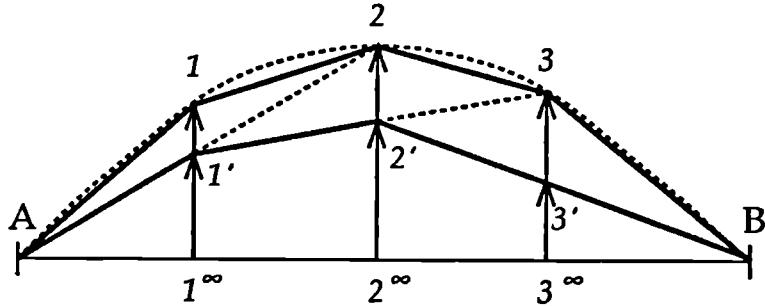
The preconditioning matrix K used here is based on the incomplete Cholesky decomposition and defined by equation (4.14), forcing L and U to have the same sparsity pattern as A .

4.2.4 Convergence Analysis - Motivation for Multigrid Acceleration

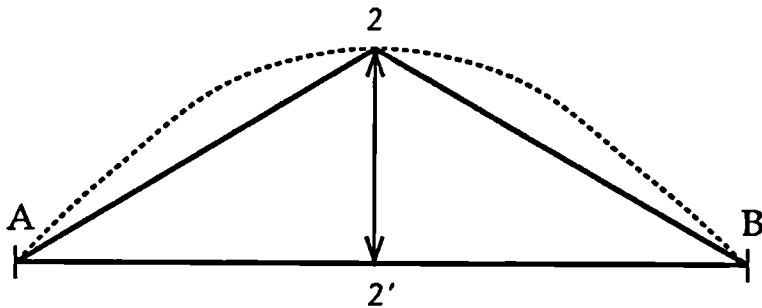
As noted earlier, direct solution of the matrix equation (4.1) results in an inverse matrix A^{-1} which is in general full, in contrast with the matrix A which is typically sparse. This reflects the fact that the value at any grid node depends on the conditions imposed at all the boundaries of the computational domain, on the distribution of the sources throughout domain and on the transport properties of the flow and the continuum, being described by the coefficient matrix A . However, as a result of the utilisation of an iterative technique, which preserves the sparsity of matrix A , the value at any considered node is influenced only by the values at the neighbouring nodes. In a similar way, these neighbour nodes are influenced by their neighbours, and so on, until finally the nodes next to the boundaries are reached and the boundary conditions are felt. Throughout this cascade process of information exchange, generated by an iteration sequence, every computational node can be regarded as a carrier of information from the boundaries, where the information received from the neighbours is passed back to them weighted according to the local transport properties and source at that location. It is quite clear that the efficiency of information exchange between two adjacent nodes is not affected by the number of nodes involved in calculation. However, due to the cascade nature of the iterative process, the efficiency of the information exchange between any node and the boundaries slows dramatically with an increase in the number of nodes involved in the calculation.

As an example of this behaviour, an elastic rope fixed at its two ends is considered. The initial shape of the rope is such that the slope of the rope is changing and this shape is controlled by a certain number of control nodes located between boundary nodes (Figure 4.1). The problem to be solved is that these control points have to be adjusted in such a way that the slope of the rope is the same on the both sides of each point. In this

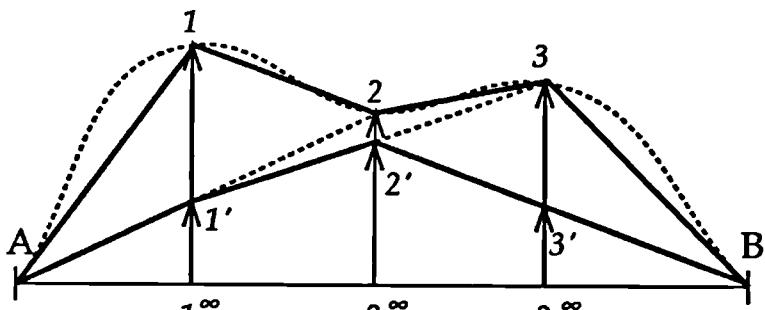
case the vertical distance from a control point to the straight line connecting the boundary nodes represents the error at that point.



a) Many iterations are needed to reduce the initial smooth error profile below the desired level



b) Smooth error profile becomes oscillatory on a coarse grid - one iteration sufficient for convergence



c) Oscillatory error profile is quickly smoothed

Figure 4.1 Rate of convergence and number of grid points

An iterative procedure equivalent to Gauss-Seidel iteration which can be used for the solution of this problem can be formulated as follows. At each iteration the new position of each control point is obtained as linear interpolation of the positions of its two neighbours (Figure 4.1), in a left to right scanning sequence. Once when the new value is obtained, it is used

to find the new position at the next control point. It is clear from Figure 4.1.a that many iterations will be needed before the error at each control point is sufficiently small. In the contrast to this situation, in Figure 4.1.b only one control point is used and only one iteration is then enough to fulfil the task. It is easy to see that any further increase in the number of control points slows convergence.

The initial oscillatory nature of the error profile shown on Figure 4.1.c disappears after one iteration, as it does in the case shown on Figure 4.1.b, and remaining profile is similar to the one presented on Figure 4.1.a. The oscillatory mode of the error has wave-length comparable to the distance between control points. Since the iterative process is efficient in exchanging the information locally, the amplitude of these error components, which have short range, are annihilated quickly, leaving the smooth error components, which span over the entire domain, to dominate the error profile. More formal discussion of this behaviour of basic iterative solvers is presented in what follows.

The computational cost or work W of an iterative algorithm is defined as the number of computational operations per iteration. In practice, computer time will, among other things (type of machine, the skill of the programmer etc.), depend on W . If N is the number of unknowns, the cost of one iteration of an iterative technique is proportional to N . The required number of iterations K and the corresponding total computational cost KW for a desired error reduction depends on the convergence factor μ_c which is defined as:

$$\mu_c = \frac{\|\varepsilon^{(k+1)}\|}{\|\varepsilon^{(k)}\|} \quad (4.23)$$

where $\|\cdot\|$ indicates a suitable vector norm. It can be shown (Varga [1962]) that μ_c tends asymptotically to the spectral radius $\rho(M)$ as $k \rightarrow \infty$. In the early stages of the classical iterative process the initial residual reduction rate is usually much higher than $\rho(M)$, slowing down gradually to the asymptotic value. The explanation for this behaviour is to be found by analysing the convergence factor for a given iterative method using the technique of '*local mode analysis*' (Brandt [1977], Stuben & Trottenberg

[1982]) in which the error $\varepsilon^{(k)}$ is expanded locally as a Fourier series. This enables one to study the way in which the given iterative method treats the different frequencies in the error spectrum.

The convergence properties of the Gauss-Seidel method applied to a simple 'model problem' will now be analysed to illustrate the above points. The conclusions will be valid with regard to the general convergence properties of iterative methods.

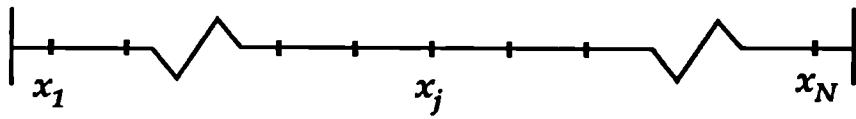


Figure 4.2 One-dimensional grid

By discretising the one dimensional Poisson's equation (Briggs [1987], Hageman & Young [1981]):

$$\frac{\partial^2 \Phi}{\partial x^2} = Q(x) \quad (4.24)$$

using central differencing on a uniform grid (Figure 4.2) with spacing h , and applying Gauss-Seidel iteration, the error equation (4.8) for point j may be written as:

$$\varepsilon_j^{(k+1)} = \frac{1}{2} (\varepsilon_{j-1}^{(k+1)} + \varepsilon_{j+1}^{(k)}) \quad (4.25)$$

Expanding the error locally as a Fourier series gives (Brandt [1977])

$$\varepsilon_j^{(k)} = \sum_{m=1}^N A_m^{(k)} e^{i \frac{m\pi}{N} j} \quad (4.26)$$

where N is the number of grid points, $i = \sqrt{-1}$, A_m is the amplitude factor, and m is the number of half sine waves which constitute $\varepsilon^{(k)}$ on the domain of the problem and is called *the wave number*. A graphical

presentation of the error components corresponding to the specific wave numbers is given in Figure 4.4. Substitution of (4.26) into (4.25) yields the ratio of amplitude factors:

$$\frac{A_m^{(k+1)}}{A_m^{(k)}} = \frac{e^{i \frac{m\pi}{N}}}{2 - e^{i \frac{m\pi}{N}}} \quad (4.27)$$

Hence, the convergence factor of the m^{th} component of the error series may be found (by taking the square root of (4.27) and multiplied by its complex conjugate) as:

$$\mu_c^m = \left| \frac{A_m^{(k+1)}}{A_m^{(k)}} \right| = \left(\frac{1}{5 - 4\cos(\frac{m\pi}{N})} \right)^{1/2} \quad (4.28)$$

The modes in the lower half of the spectrum, with wavenumbers in the range $1 \leq m \leq N/2$, are called *low-frequency* or *smooth* modes (Figure 4.4.a). The modes in the upper half of the spectrum, with $N/2 < m \leq N$ are called *high-frequency* or *oscillatory* modes (Figure 4.4.b). A quick look at Figure 4.3 tells us that the error clearly decreases with each iteration but the rate of decrease is larger for the higher wavenumbers. This is consistent with the experience of many practitioners, who found that the majority of basic relaxation schemes work well during the initial process of iteration: however, soon afterward the convergence slows down and the entire scheme appears to stall. The spectral (or Fourier mode) picture of the Gauss-Seidel relaxation scheme applied to the model problem above explains that the rapid decrease in error during the early iterations is due to the efficient elimination of the oscillatory modes of the error. Once they have been removed, iteration is much less effective in reducing the remaining smooth components. Many iteration schemes possess this property of eliminating the oscillatory modes and leaving the smooth ones: this is called the *smoothing property*. It is a serious limitation of these methods.

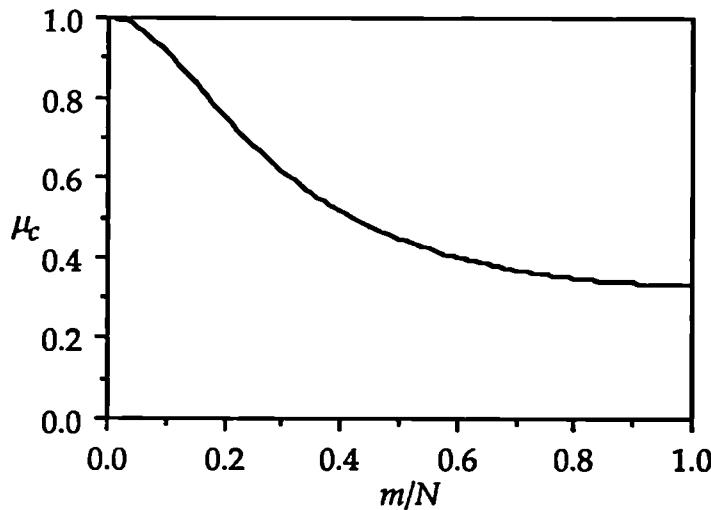


Figure 4.3 Distribution of eigenvalues according to equation (4.28)

The largest value of μ_c^m , which governs the asymptotic rate of convergence (Hageman & Young [1981]), occurs at the smallest value of the wavenumber, $m=1$ (Figure 4.3). In the limit, as k tends to infinity, equation (4.28) shows that on a grid of 40 computational points, the spectral radius of the Gauss-Seidel iteration matrix is 0.994, so that 375 iterations are required to reduce the error by an order of magnitude, according to equation (4.13). At the same time the largest convergence factor μ_c in the low-frequency spectrum corresponds to $m=N/2$ and its value is 0.45 (equation 4.28), which means that, irrespective of grid spacing, the high wavenumber error components are reduced by an order of magnitude in just 3 iterations.

Any attempt to improve the accuracy of the solution by reducing the grid spacing will only worsen the convergence of the smooth components of the error, increasing the number of iterations K necessary to reduce error below a prescribed level. The result of this is that the total computational cost $W=O(N^p)$ where $p>1$. Practical experience indicates that the estimates for the total computational cost derived for Poisson's equation (GS: $p=2$, SOR: $p=1.5$) holds for a wider class of equations. For the preconditioned CG solver applied to the two-dimensional Poisson's equation Gustaffson [1978] has shown the theoretical value of p to be 1.25.

4.3 MULTIGRID PHILOSOPHY

In order for an error component to be visible on a certain grid, a contribution from every peak and trough must be representable on that grid. On a 'fine' numerical grid with uniform mesh spacing h and N computational points, it is possible to present the error modes in the range $1 \leq m \leq N$ ($N=8$ for case in Figure 4.4). If a 'coarse' grid is defined with mesh spacing $2h$, it can be seen from Figure 4.4 that the high-frequency error components are only 'visible' (Stuben & Trottenberg [1982]) on the 'fine' grid (Phillips [1984]). However, it was shown in the previous section that the oscillatory error components are annihilated in very few iterations, irrespective of grid spacing. The basic multigrid idea is to exploit the latter property by realising that smooth modes on a fine grid look less smooth on a coarse grid. Note in Figure 4.4 that the $m=8$ mode for the fine grid is equivalent to the $m=4$ mode for the coarse grid. Thus, by suitably approximating the error on a series of progressively coarser grids, each Fourier component can be efficiently reduced using a grid on which it has a short wavelength. Since the reduction of short-wavelength errors does not depend on the grid spacing, the multigrid convergence rate is independent of grid spacing, and the total work varies linearly with the number of grid points.

Fodorenko [1962] was the first to suggest the multigrid idea, and the theoretical optimality has been established many times for model problems, but it was Brandt [1977] who developed the first practical multigrid algorithms.

The MG method may therefore be summarised as one which iteratively solves a system of discrete equations on a given grid, by continual interactions with a hierarchy of coarser grids, taking advantage of the relation between the different discretisations of the same continuous problem. This method can be viewed in two complementary ways (Brandt [1977]): one is to view the coarser grids as correction grids, accelerating convergence of a relaxation scheme on the finest grid by efficiently removing smooth error components. The other is to regard

finer grids as the correction grids, improving solution accuracy on coarser grids by correcting their forcing terms.

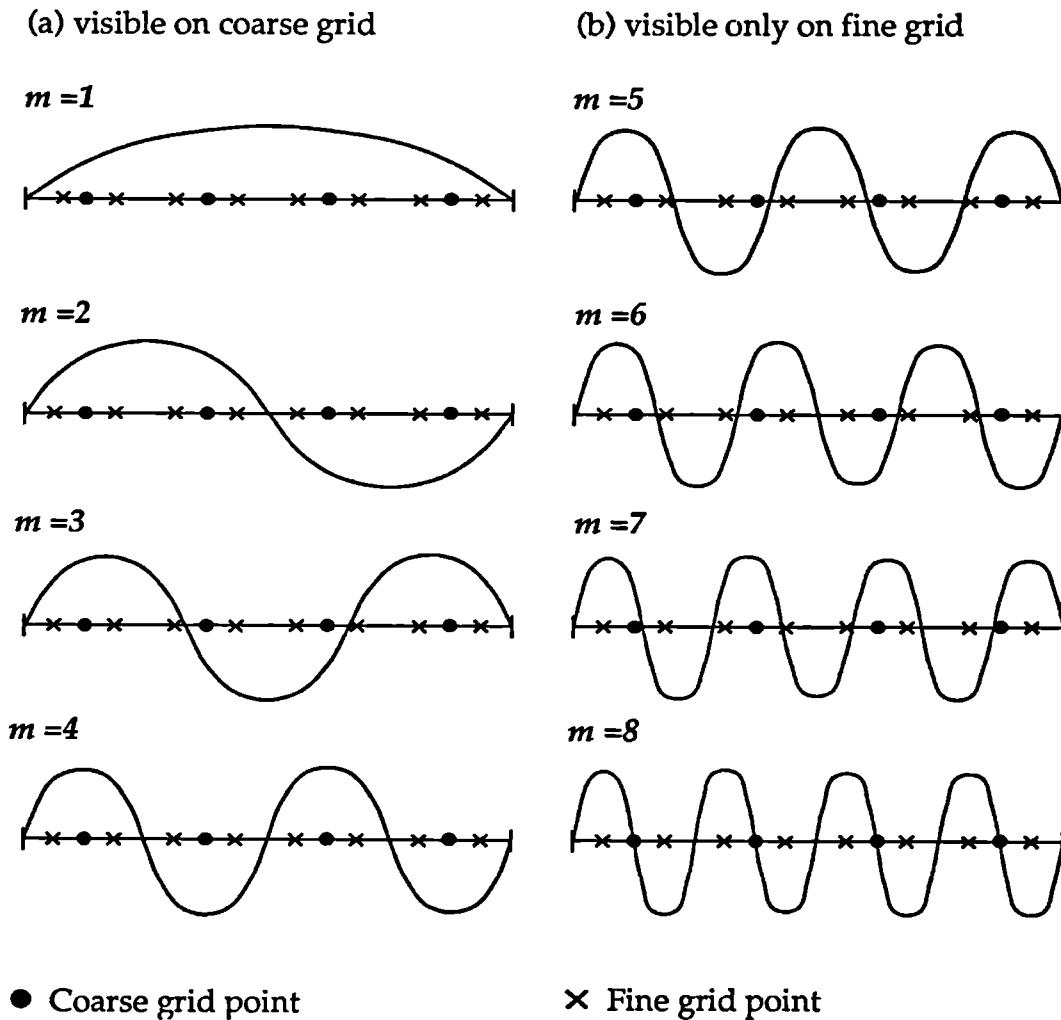


Figure 4.4 Components of $\sin(\frac{m\pi}{Nh})$ visible on coarse and fine grids

In order to represent the error on coarser grids, a coarse grid version of the error equation (4.8) must be derived. The standard coarse grid discretisation of the problem is not suitable since it is not a true representation of the fine grid equation. This is because the accuracy of the discretised equation depends on the mesh spacing, so that the coarse and the fine grid solution would converge to different values. Instead, a modified coarse grid equation is required, which exactly represents the fine

grid error as the solution proceeds (Brandt [1977]). The manner of doing this will be explained later.

The coarse grid equation can be written in terms of corrections to the fine grid solution, which are eventually added to it. The broad category of MG methods, designed to work on coarse grids with corrections only, is called Correction Scheme (CS), and is suitable for linear systems. For coupled systems of strongly nonlinear algebraic equations, it is more appropriate to devise schemes which operate on the solution (rather than its correction) on all grid levels, such that coefficients and sources can be updated as the solution advances. This class of MG techniques is called Full Approximation Storage (FAS) scheme and, clearly, is the appropriate method to adopt in the case of the equations encountered in fluid mechanics.

Generally, there is no particular difficulty in applying the MG to a single equation, or a system of linear equations on a regular grid. However, a considerable challenge is presented by the highly coupled and complex sets of governing equations describing practical laminar or turbulent flows. The details of the MG implementation are closely associated with the fundamental solution algorithm used to couple the various sets of algebraic equations, each pertaining to a related flow variable. Most schemes for turbulent recirculating flows sequentially solve the sets of algebraic equations arising from the discretisation process of the related governing equations, because of simplicity, flexibility and low-storage requirements. In these circumstances, the resulting solution convergence rate can be seriously affected if the coupling of the various governing equations is not properly honoured. The performance of a MG method depends crucially on the manner in which the MG is integrated with the overall iterative scheme.

Within an uncoupled or segregated algorithm, the most straightforward implementation of the MG method is to apply it to each non-linear set separately, without taking any account of inter-set coupling. Such a simple approach was adopted, for example, by Phillips et al. [1985] in a calculation of a turbulent recirculating flow behind a backward-facing step with the $k-\epsilon$ eddy-viscosity model. In this case, only a very modest

saving in CPU time was achieved, due to the uncoupled nature of their MG implementation. Sivaloganathan and Shaw [1988] recognised the above deficiency and instead chose to embed the SIMPLE algorithm as a smoother in their FAS-MG method. The latter implementation is used in the present study.

Convergence analysis for multigrid methods is difficult to perform. It is an area which is still filled with dark corners and unsolved problems. Heuristic and qualitative arguments suggest that the standard multigrid schemes, when applied to well-behaved problems (for example, systems of linear algebraic equations with symmetric positive definite coefficient matrices) work very effectively. Convergence of such problems can be proved quite rigorously. Where analytical results are lacking, a wealth of computational evidence testifies to the general effectiveness of multigrid methods. However, multigrid convergence analysis is still an open area of computational mathematics.

4.4 COARSE-GRID EQUATIONS

The multigrid concept does not impose any limitation on the nature of the discrete system to be solved, and as such it can be used, not only to speed up the convergence of the inner-iteration sequence performed on the linearised system, but also to construct multigrid algorithms which operate on the outer iterations as well. This second form of implementation also helps to resolve non-linearities and enhance inter-equation coupling of a segregated algorithm (see Section 3.6).

Discretisation of the governing equations on a coarser grid results in an approximation of the fine-grid algebraic system. The solution obtained on the coarse grid may be used as an initial guess on the fine grid. This idea of using coarser-grid results to generate improved initial guesses on finer grids is the basis of a strategy called *nested iteration*. In general, however, the solution on a coarser grid does not satisfy the fine grid-algebraic system, since these two systems are obtained by different discretisations

and will therefore converge to different solutions. The object of this section is to describe a way of representing the fine grid algebraic system on a coarser grid, so that, once the iterations stall on the fine grid due to the presence of smooth error components, a coarse grid formulation of the original system may be used for their efficient elimination, as described in the previous section.

In Section 3.5 the final form of discretised equations was given. The procedure in which these coupled and strongly nonlinear equation sets were to be solved within the SIMPLE algorithm was described in Section 3.8. The exact solution of the discretised equations for variable $\phi = (v_i, p, T, k, \epsilon, \dots)$ on a fine grid f satisfies the following equation:

$$A_f \phi_f = b_f \quad (4.29)$$

where the subscript f is used to denote the equation to be solved on the fine grid (Figure 4.5). A_f and b_f are the exact coefficient matrix and source term vector, being functions of the solution, i.e. $A_f = A_f(\phi_f)$ and $b_f = b_f(\phi_f)$.

After several outer iterations on the whole set of equations (as explained in Section 3.8), an approximate solution ϕ_f^* is obtained, which satisfies equation (4.29) only to a residual r_f :

$$A_f^* \phi_f^* = b_f^* - r_f \quad (4.30)$$

:

where A_f^* and b_f^* are approximations to A_f and b_f based on the approximate solutions ϕ_f^* . Subtracting (4.30) from (4.29) yields:

$$A_f \phi_f = b_f + A_f^* \phi_f^* - b_f^* + r_f \quad (4.31)$$

When the equations are linear, i.e. such that A and b are independent of the solution and given as known values, equation (4.31) can be written in terms of the correction $\phi_f' = \phi_f - \phi_f^*$:

$$A_f \phi_f' = r_f \quad (4.32)$$

This indicates that the residual of equation (4.32) acts as the source of correction needed to bring ϕ^* to match ϕ , as was noted in Section 4.2.1, where linear stationary iterative techniques were discussed. In Sections 4.2.4 and 4.3 it was indicated that it might be beneficial to solve this equation partly on a coarser grid, for this is cheaper and usually helps to keep a nearly constant rate of convergence. Equation (4.32) is used to formulate the coarse-grid equation for the Correction Scheme multigrid method.

For nonlinear equation systems, the correction ϕ' depends not only on the value of the residual r_f but also on the changes of sources and coefficients, which are, as mentioned above, functions of the advancing solution ϕ^* . If one writes (Peric [1990]):

$$\begin{aligned} A_f &= A_f^* + A_f' \\ \phi_f &= \phi_f^* + \phi_f' \\ b_f &= b_f^* + b_f' \end{aligned} \quad (4.33)$$

equation (4.31) reduces to:

$$A_f^* \phi_f' = r_f + b_f' - A_f' \phi_f^* - A_f^* \phi_f' \quad (4.34)$$

showing additional dependence terms not present in equation (4.32). Although equation (4.32) can be solved for ϕ' , equation (4.34) cannot, since b_f' and A_f' are not, in general, linear functions of ϕ , i.e. $b(\phi + \phi') \neq b(\phi) + b(\phi')$. In laminar flows, the only b -term is the pressure source, which is linear. Barcus et al. [1987] used a Correction Scheme to solve equation (4.34) for ϕ' for the nonlinear problem, thereby neglecting the last two terms on the right hand side.

In order to keep track of all nonlinear dependencies and intervariable couplings, equation (4.31) is used as the basis for multigrid solution in the full approximation (FAS) scheme. Fine-grid quantities are interpolated (restricted) to the coarse grid c (Figure 4.5), as described in Section 4.6.2. These restricted quantities are denoted by a tilda superscript \sim . An approximation to equation (4.31) on the coarse grid leads to the *coarse grid*

equation (4.35) for ϕ , whose solution is ϕ_c and the changes $\phi'_c = (\phi_c - \tilde{\phi}_c)$ are driven by the fine grid residual (Brandt [1977], Smith [1990]) i.e..

$$A_c \phi_c = b_c + \underline{\tilde{A}_c \tilde{\phi}_c} - \underline{\tilde{b}_c} + \underline{\tilde{r}_c} \quad (4.35)$$

or in terms of an approximate solution on the coarse grid ϕ_c^* :

$$A_c^* \phi_c^* = b_c^* - r_c \quad (4.36)$$

Here A_c , ϕ_c and b_c are approximations to A_f , ϕ_f and b_f respectively, $\tilde{\phi}_c$ and \tilde{r}_c are the fine grid solution ϕ_f^* and residual r_f vectors restricted (as will be explained later in Section 4.6.2) to the coarse grid. \tilde{A}_c and \tilde{b}_c are the coefficient and source, calculated on the coarse grid in the same way as A_f^* and b_f^* are calculated on the fine grid but now based on the restricted fine grid solution $\tilde{\phi}_c$. The meanings of the terms in (4.36) are identical to those of (4.30), but now all terms are related to the coarse grid and the underlined terms of equation (4.35) are lumped into b_c^* . It is useful to observe that the underlined terms in equation (4.35) are fixed during coarse-grid iterations.

Initially ϕ_c^* is equal to $\tilde{\phi}_c$, and, therefore, $A_c^* = \tilde{A}_c$ and $b_c^* = \tilde{b}_c$. If the residuals were absent (i.e. fine grid solution ϕ_f^* already equal to ϕ_f), than equation (4.35) would simply result in $\phi_c^* = \tilde{\phi}_c$, i.e. the restricted variable values will remain unchanged and will satisfy the coarse grid equation identically. When the residuals are nonzero, ϕ_c^* will depart from $\tilde{\phi}_c$ as the SIMPLE smoothing scheme is applied to the coarse-grid equations, and accordingly A_c^* and b_c^* will depart from \tilde{A}_c and \tilde{b}_c .

All the transport equations (v_i , T , k , ϵ) have the same coefficients and sources as in the single-grid case, described in Chapter 3. The only difference is an additional source (the underlined term in equation (4.35)) added to the coarse-grid equation to give fine-grid accuracy to the coarse-grid solution. Although we are solving on the coarse grid for ϕ_c , which is an approximation to the exact fine grid solution ϕ_f , the coarse grid equation (4.35) should be seen as an equation for variable correction ϕ'_f .

Indeed, equation (4.35) is nothing else but equation (4.34), only in another form.

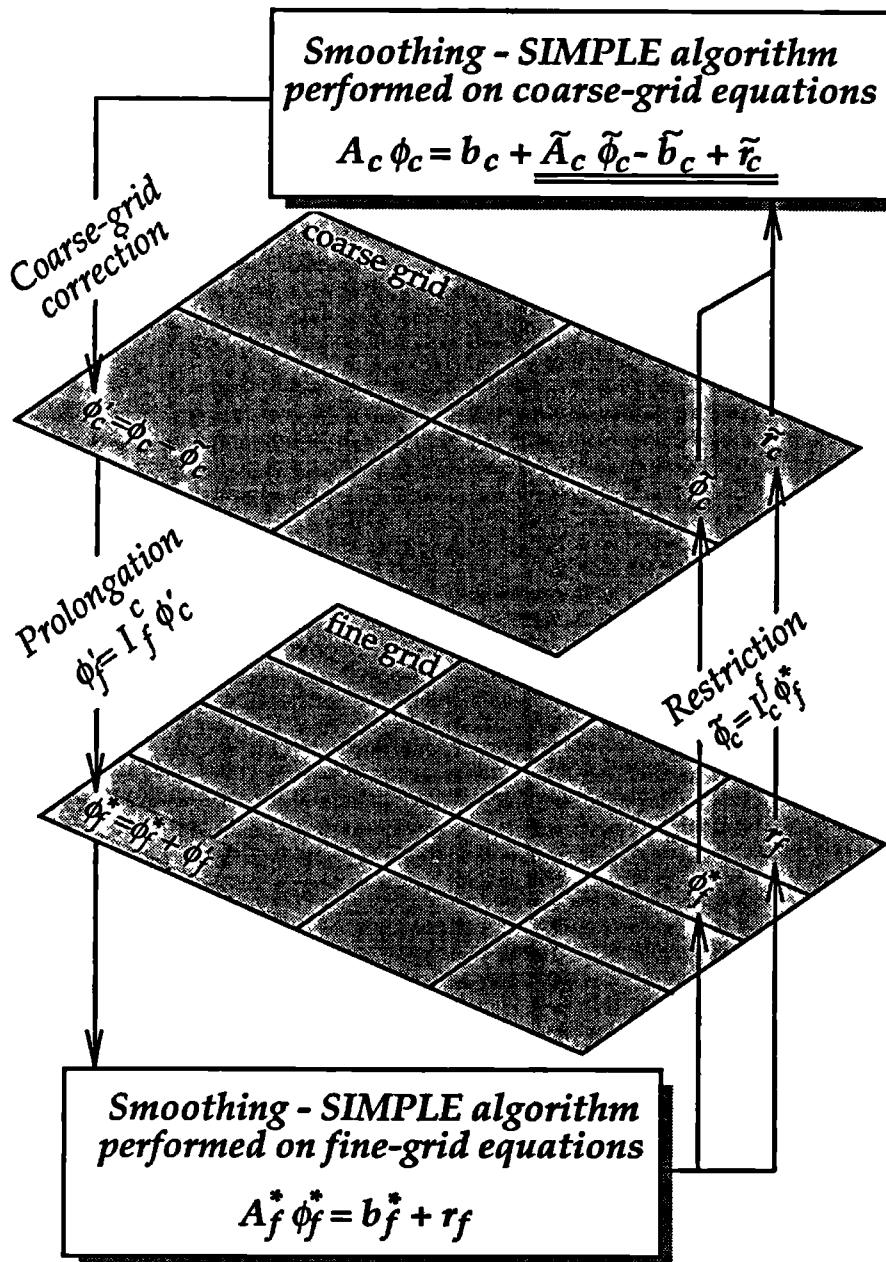


Figure 4.5 Fine- and coarse-grid multigrid coupling

The SIMPLE algorithm is linear in pressure and operates with pressure corrections. We, therefore, do not need to restrict pressure from fine to coarse grid. Instead, we can work on the coarse grid with pressure correction p^* (Peric [1990]), which is initially equal zero.

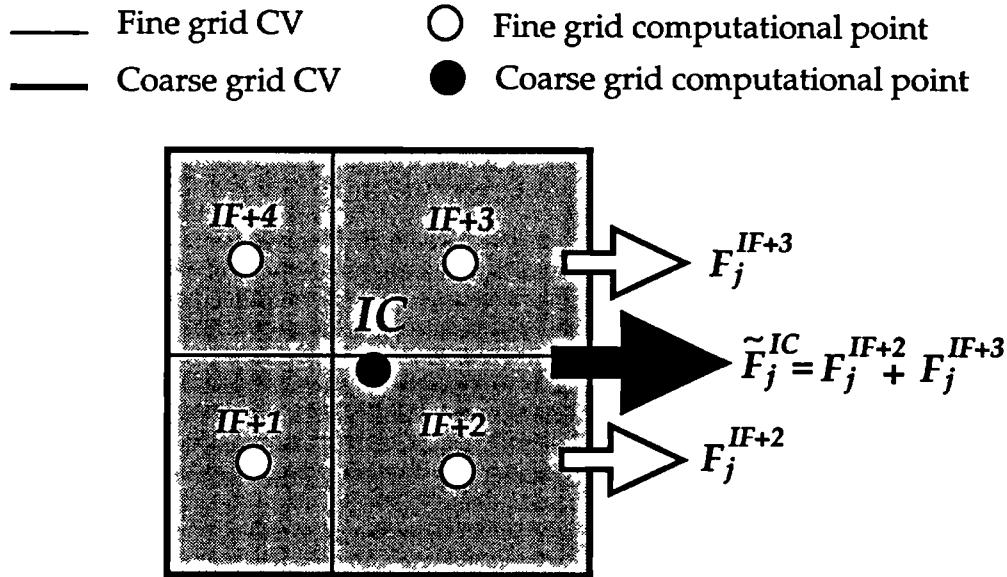


Figure 4.6 Fine and coarse grid arrangement

The coarse grid mass flux should not be calculated from the restricted velocity field $\tilde{\underline{v}}$, since it cannot be assured that the values of $\tilde{\underline{v}}$ interpolated to the cell face give mass fluxes that satisfy the continuity equation locally, even if the fine grid solution were 'exact'. Instead, the initial mass fluxes \tilde{F}_j on the coarse grid are conservatively evaluated by summing the fine grid mass fluxes through the corresponding cell faces (Figure 4.6). Thus, when the fine grid mass fluxes satisfy the continuity equation, this is also automatically ensured on the coarse grid. During the coarse grid iteration, the change of these initial fluxes are driven by pressure correction p^* and changes in velocity, such that expression (3.29) becomes:

$$F_j^* = \tilde{F}_j + \rho_j \left(\frac{\underline{v}_{Poj}^* - \tilde{\underline{v}}_{Poj}}{K_j} + K_j \left\{ \frac{1}{2} [(\nabla p_{P_o}^*)_{d_j} + (\nabla p_{P_j}^*)_{d_j}] - \tilde{\nabla p}^* \right\} \frac{S_j}{|S_j|} \right) S_j \quad (4.37)$$

The pressure-correction equation has the same coefficients as in the single-grid case (Section 3.6.3). The only difference is the calculation of mass fluxes, which is now based on equation (4.37). In the converged solution $F_j^* = \tilde{F}_j$, and hence $p^* = 0$.

After a few outer iterations (Section 3.8) performed on the coarse grid system (4.36), the coarse-grid corrections ϕ_c' are calculated (Figure 4.5) and interpolated back (prolongated) to the fine grid to adjust the fine-grid solution (a detailed descriptions of the prolongation operator is given in Section 4.6.3). Due to interpolation some high-frequency errors are thereby introduced. However, these errors are eliminated in a few fine grid iterations. Since the updated solution on the fine grid will still not, in general, be converged, it is necessary to repeat this two-level cycle until full convergence is achieved.

What has been described here is, strictly, a two-grid method. However, equation (4.36), being of the same form as the original equation (4.30), may itself be solved using a coarser grid, so that the extension of the solution procedure from two-grid scheme to an arbitrary multigrid scheme is straightforward.

4.5 COARSE-GRID BOUNDARY CONDITIONS

The FAS coarse grid finite volume equations may be thought of as the standard coarse-grid equations modified to give the fine-grid accuracy as the solution proceeds. This means that the same dependent variable is solved for on both coarse and fine grids, so that the coarse-grid boundary conditions are implemented in the same way as the fine ones, as described in Section 3.7.

4.6 COMPONENTS OF A MULTIGRID SOLVER

In order to fully define a multigrid algorithm, the following choices remain to be made:

- (1) Coarse and fine grid arrangement.

- (2) Restriction operator.
- (3) Prolongation operator.
- (4) Iterative scheme used in inner iteration sequence.
- (5) Multigrid cycle.

These will now be separately considered.

4.6.1 Coarse and Fine Grid Arrangement

Multigrid theory does not assume any particular relation between the space discretisation on the coarse and fine grids. However, based on local mode analysis (Section 4.3) for model problems, Brandt [1977] recommends that the coarse grid mesh spacing should be about twice that of the next fine grid to achieve maximum efficiency. This is relatively easy to achieve by either amalgamating groups of fine grid cells, producing progressively coarser meshes or by subdividing cells in some manner in order to obtain the finer meshes. However, when unstructured meshes are used, the construction of coarser grids by amalgamating groups of fine grid cells does not enable good control of the cell topology, since it is difficult to predict all possible situations which may occur during this process. This can adversely affect the stability and accuracy of calculations on the coarser grids (Section 3.2). Although this does not affect the accuracy on the finest grid, it can hinder convergence of the multigrid algorithm. Generation of fine meshes by repeatedly subdividing cells of a coarse unstructured mesh gives much greater control, since the possible combinations are limited by the number of different cell topologies used to generate the coarser grid and the strategies that are used for subdividing them. Furthermore, this approach is well suited for adaptive local grid refinement (Chapter 5) and the full multigrid cycle (Section 4.6.5).

At the initial stage of a calculation only a coarsest grid must be generated, containing only the minimum cells needed to describe the general features of the original geometry. All finer meshes can then be generated automatically by subdividing the coarser grids and adjusting the boundary control volumes to fit the boundary surface.

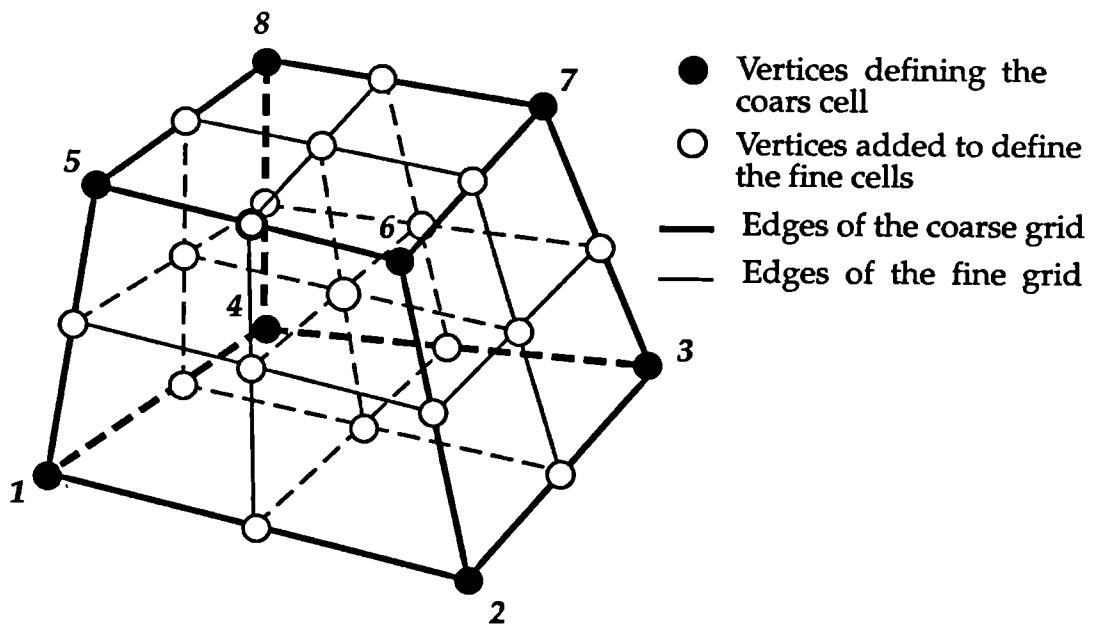


Figure 4.7 The construction of the fine grid

In this study the fine grid cells are constructed by subdividing a coarse cell into eight (four in 2D calculations) cells, as shown in Figure 4.7 (It is also possible to subdivided only in selected directions, but this option is not adopted here). As a result, the faces of the coarse cell are coincident with those of the fine cells derived from the coarse one (the so called 'coincident cell face arrangement' (Smith [1990])). Summing the discretised equations of the fine cells comprising the coarse cell results in the coarse grid equation (4.35) (Smith [1990]). For the hexahedral control volumes presented in Figure 4.7 this procedure is unique and preserves the topological properties of the coarse cell. However, specific subdivision methodologies must be developed for every cell topology used in a spatial discretisation. This aspect will not be discussed further in the present work.

4.6.2 Restriction Operator

The transfer of field values defined on the fine grid to the next coarser one is called 'restriction'; and the operator which denotes this transfer is called the restriction operator I_c^f :

$$\tilde{\phi}_c = I_c^f \phi_f^* \quad (4.38)$$

According to Brandt [1977], the restriction operator for problems with rapidly varying coefficients of the system of algebraic equations (typical in flow calculations) should be an average of all fine-grid computational points surrounding the coarse grid one, rather than simple 'injection' in which the value of the coarse grid point is taken as the value at the coincident fine grid point. This is supported by the numerical results of Ghia et al. [1982] for the well-studied driven cavity flow, who, using a finite-difference approximation, reported improved convergence using a weighted operator rather than injection.

For the coarse-fine grid arrangement adopted in the present study, shown in Figure 4.6, injection is not applicable (because the node of the coarse cell does not coincide with any fine-cell nodes), and the following linearly weighted restriction operator (Smith [1990]) is selected:

$$\tilde{\phi}_{c,IC} = \frac{1}{nc} \sum_{i=1}^{nc} \phi_{f,IF+i}^* \quad (4.39)$$

where IC is the index of the coarse-grid cell, nc the number of fine cells comprising the coarse one, and IF is a pointer which relates the coarse cell to the fine ones it contains (Figure 4.6).

Since the residuals r_f and \tilde{r}_c that appear in the expressions (4.30) and (4.35) are not functions of space and time coordinates only (i.e. they are not strictly fields), but also depend on the mesh size, interpolation based on their space variation is inappropriate. The residual of the coarse grid \tilde{r}_c is therefore obtained by simple summation of residuals of the fine cells comprising the coarse cell (Figure 4.6), i.e.

$$\tilde{r}_{c,IC} = \sum_{i=1}^{nc} r_{f,IF+i} \quad (4.40)$$

4.6.3 Prolongation Operator

The prolongation operator I_f^c , as introduced in Section 4.4, transfers field values from the coarse to the fine grid:

$$\phi_f = I_f^c \phi_c \quad (4.41)$$

A prolongation operator may represent a simple transfer of values to the fine grid assuming that the field values are constant over a coarser grid cell (Smith [1990]), or it may represent transfer of some weighted averages. Brandt [1977] states that the order of interpolation should be no less than the order of the parent differential equations. Thompson and Ferziger [1989] implemented two-dimensional Lagrange interpolation with four, nine and sixteen points for both the prolongation and restriction operators. Their tests indicate that it is especially bad to use high-order interpolation for the coarse to fine grid transfer. This observation was explained by the possible introduction of high-frequency noise that the fine grid iterations had to remove. The four-point Lagrange interpolation (i.e. equivalent to linear interpolation) was finally selected by them as the optimum.

In this study, tests have been done with the simple assumption that the value of a field variable is constant within a coarse grid cell (Smith [1990]):

$$\phi_{f,IF+i} = \phi_{c,IC} \quad i=1,\dots,nc \quad (4.42)$$

and by linear interpolation based on expressions (3.10) and (3.11):

$$\phi_{f,IF+i} = \phi_{c,IC} + (\nabla\phi)_{c,IC} \cdot (x_{IF+i} - x_{IC}) \quad i=1,\dots,nc \quad (4.43)$$

The influence of these practices on the convergence properties of the multigrid algorithm is assessed and discussed in Chapter 6.

4.6.4 Iterative Scheme Used in Inner Iteration Sequence

As already pointed out (Section 4.3), in a multigrid method, the role of the inner iterative relaxation scheme is not so much to reduce the residuals as to smooth their distribution, by eliminating the high-frequency error components. Although local mode analysis can be used to determine the most suitable relaxation operator for some model problems (Brandt [1977]), practical engineering problems are usually too complicated to be analysed with this approach, and the best choice can be found only by trial and error. Furthermore, the relaxation techniques described in Section 4.2.1 are devised for systems of linear algebraic equations. When these techniques are implemented in a SIMPLE-like (Section 3.6.3) overall iterative algorithm, designed to handle systems of nonlinear coupled algebraic equations, the convergence properties of the overall algorithm depend, not only on the rate of convergence of the inner iteration sequences, but also on the nature of nonlinearity and inter-equation coupling, as well as on the overall algorithm itself. This is due to the fact that during outer iteration the intermediate solutions of the linearised algebraic system are approximations of the converged solution. This is equally true if within each outer iteration the residual vector norms of the linearised systems are reduced to the value limited by the computer accuracy or just by a fraction of its initial value. Experience shows that efforts to converge the inner solutions to a very tight tolerance do not pay back in speeding up the overall algorithm. Any stable and convergent relaxation method for the solution of the system of linearised algebraic equations can be used without seeing significant reduction in the overall number of outer iterations.

In this study, the Gauss-Seidel, incomplete LU decomposition, and incomplete Cholesky preconditioned bi-CG relaxation operators are used for solution of all the discretised transport equations, except for the continuity equation, and their relative performance is evaluated. The preconditioned CG solver is used for solution of the pressure correction equation (Section 3.6). Although the method is not a good smoother in the sense of Section 4.2.4., it is chosen because the pressure-correction equation does not exhibit diagonal dominance, since the central coefficient is always equal to the negative sum of the neighbour coefficients. When classical

relaxation methods are applied to the pressure-correction equation, they consequently often converge very slowly or diverge. In addition, it is usually important to secure mass conservation at each outer iteration, which necessitates solving the pressure correction equation (3.37) to a tight tolerance.

4.6.5 Multigrid Cycle

The coarse grid equations used in the multigrid process are only approximate (until convergence is reached), so that there is no incentive to solve them exactly. On the other hand, it is important to ensure that the long wavelength error components are sufficiently reduced on the coarse grid so that they do not dominate the convergence rate after prolongation to the fine grid.

The order in which the grids of different resolution are visited defines the multigrid cycle, the aim of which is to ensure that each grid is used to its full potential. Many different variations are possible, and V, W and Full Multigrid (FMG) V cycles are shown in Figure 4.8. Figure 4.8.a shows the schedule for the grids in the order in which they are visited for V multigrid cycle. A two-level MG telescopes down to the coarsest grid, and then works its way back to the finest grid. Because of the pattern of this diagram, this algorithm is called the V-cycle. It has a compact recursive definition, which is given by:

- (1) Smoothing: perform outer iteration on the fine grid system of equations (Section 3.8).
- (2) Restriction: transfer information to the coarser grid (Section 4.6.2), and formulate the coarse grid equations (4.33).
- (3) Smoothing: perform outer iteration on the coarse grid system of equations.
- (4) Repeat (2) and (3) until the coarsest grid has been reached.
- (5) Prolongation: calculate the coarse grid correction and transfer it to the next finer grid (Section 4.6.3), and obtain a new estimate on this grid.
- (6) Smoothing: improve this estimate by performing outer iterations on the fine grid system of equations.

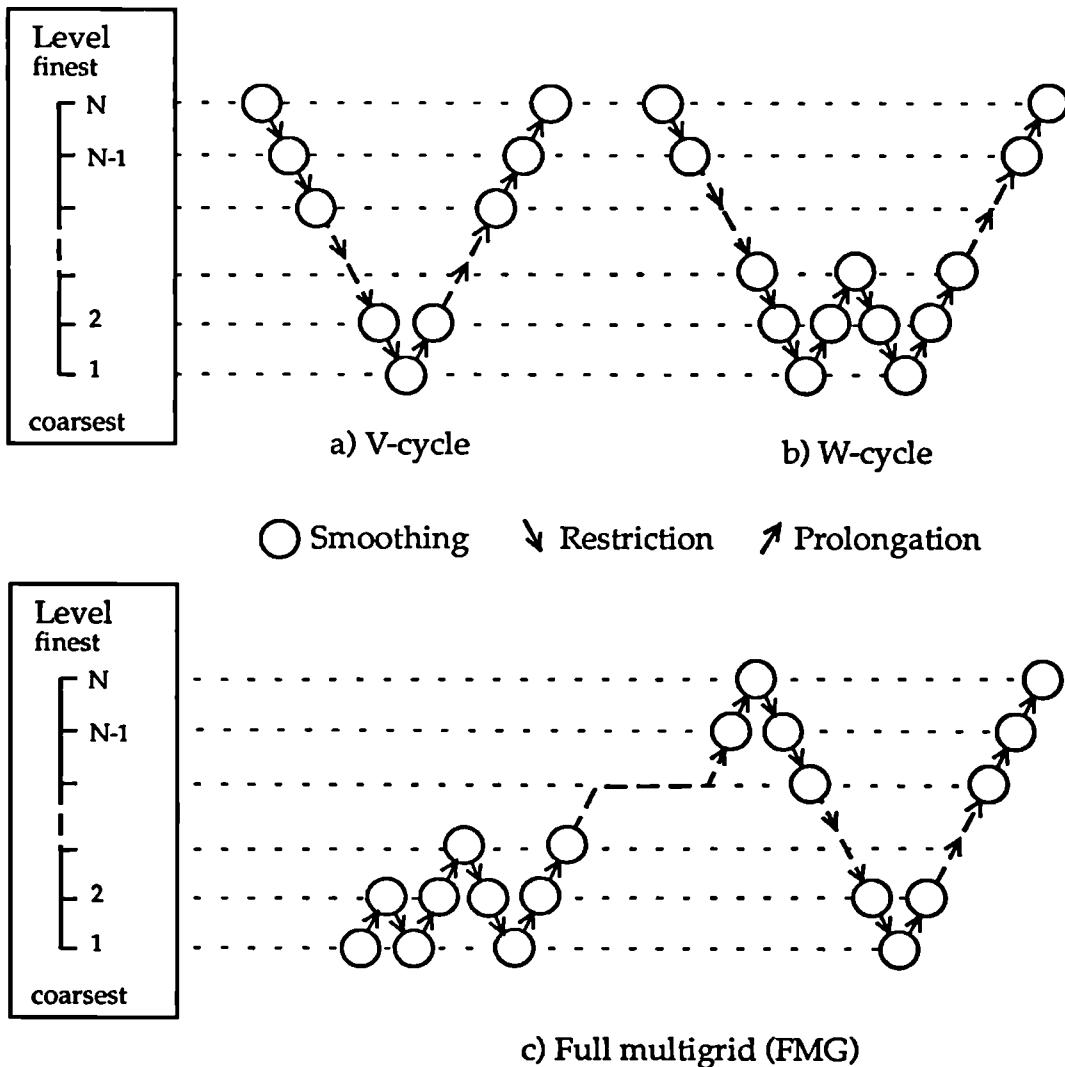


Figure 4.8 Schedule of grids for a) V-cycle, b) W-cycle and c) FMG scheme

- (7) Repeat (5) and (6) until the finest grid has been reached.
- (8) Repeat (1)-(7) until convergence criterion is satisfied on the finest grid.

The use of the nested iteration idea, which uses coarse grids to obtain improved initial guesses for a fine grid problem, leads to the full multigrid cycle, which is a remarkable synthesis of ideas and techniques that individually have been well known and used for a long time, but taken alone sometimes have serious defects. Full multigrid is a technique for integrating these techniques so that they can work together in a way that removes their limitations. The result is a simple but powerful algorithm.

Figure 4.8.c shows the FMG V-cycle, which is adopted in the present study. The algorithm starts on the coarsest grid, produces converged solution on that grid, interpolates that solution to the next finer grid and performs V-cycle on those two grids until a converged solution is achieved (Figure 4.8.c). This solution is interpolated to the next finer grid and used as an initial guess for the currently finest, third-grid level. V-cycle, as explained above, is performed on these three grids until convergence is achieved. This procedure is recursively repeated until a converged solution is obtained on the finest grid. In this way, each V-cycle is preceded by a smaller V-cycle designed to provide the best possible initial guess. The extra work done in these preliminary cycles is not only inexpensive, but generally pays for itself. Although Brandt [1980] claims that just one execution of his full multigrid cycle is sufficient to drive the solution error below the truncation error for some problems, he does not support this with numerical results.

The full multigrid V-cycle has a structure which is very natural for adaptive local grid refinement, since it introduces gradually finer grids, which can be constructed in an adaptive manner (Chapter 5).

The choice of the number of outer iterations to be performed on each grid is not obvious. There are two common approaches for the selection of these parameters:

- the '*fixed*' scheme, in which the cycling parameters are chosen *a priori* and remain fixed throughout the course of the algorithm. The choice can be made on the basis of analysis or *prior* experimentation.
- the '*accommodative*' scheme, in which the cycling parameters are determined 'on the run' to account for variations in the pattern of convergence.

For practical engineering calculations, the best specification of cycling parameters may usually be found by trial and error. The outer iteration sequence of the *accommodative algorithms* are dynamically controlled (Brandt [1977]) by the ratio of the current residual r_l^k to the previous one

r_l^{k-1} at the l -grid level, where the k defines the outer iteration counter on that grid. If

$$\frac{|r_l^k|}{|r_l^{k-1}|} < \Omega \quad (4.44)$$

Ω being a convergence-rate criterion, the relaxation scheme is deemed efficient and is continued on the same grid-level until the convergence criterion $r_l^{k+1} < v^l$ on that grid is reached, at which stage relaxation is terminated. If, on the other hand (4.44) is not satisfied, the rate of convergence is deemed to be too slow, and iterations on this grid are temporarily halted. The residuals and the solutions are then restricted to the next coarser grid-level $l-1$, and the convergence criterion of that grid v^{l-1} , is reset on the basis of the last global residual on the l -grid level

$$v^{l-1} < C |r_l^k| \quad (4.45)$$

The values for Ω and C , which give the optimum performance, may be found only by trial and error, and are typically between 0.3 and 0.6 (Lien [1992]). Accommodative algorithms are found not to perform better than the fixed ones (Lien [1992]) so in the present study the fixed approach has been employed.

4.7 CLOSURE

In this chapter, the multigrid theory for the solution of sets of simultaneous nonlinear algebraic equations has been examined. Firstly, some well-established iterative solvers were reviewed, and their drawbacks highlighted. Then, by analysing the Gauss-Seidel method applied to a simple model problem, it was shown that the slow asymptotic convergence rate and poor efficiency on fine grids of the solver is caused by the very slow decay of long-wavelength error components. It was also

demonstrated that short-wavelength components are annihilated very rapidly, independent of grid size.

The multigrid idea of using coarser grids to reduce the long wavelength error components, recognising that a long wavelength on one grid becomes shorter on a coarser grid, was then introduced. It was explained that the multigrid method requires that the fine grid residual may be suitably approximated on coarser grids, and the derivation of an appropriate coarse grid equation was examined. Finally, a complete multigrid algorithm was summarised, its individual components examined and choice made for its implementation in the present study. In the next chapter, the idea of combining multigrid and local grid refinement, in order to efficiently resolve regions of large numerical error, will be examined.

Chapter 5

Local Grid Refinement and Adaptive Mesh Generation

5.1 INTRODUCTION

Chapters 3 and 4 presented the numerical tools that enable us to obtain a discrete description of the continuous variation of the dependent variables in space, which obey the physical laws expressed by governing equations described in Chapter 2. The quality of those numerical results depends on of validity of the mathematical models used in describing the flow phenomena as well as the quality of the numerical approximations (discretisation) of the equations characterising the mathematical model. This study deals with the second issue, posing the question: how accurately does the numerical solution represent the set of governing integral (differential) equations comprising the mathematical model? To answer this question some error estimates should be available so that plausible-looking results are not confused with accurate results and the numerical error can be distinguished from the influence of modelling assumptions. The numerical error e is defined as the difference between the exact solution of the governing equation $\Phi(x,t)$ (Section 2.3) and the solution of the discrete equation ϕ obtained on a computational mesh with characteristic size h :

$$e = \Phi - \phi \quad (5.1)$$

In general, the only data available to the analyst that can give some indication of the numerical error is the approximate solution itself. Thus,

it is of interest to obtain an *a posteriori* (after an initial approximate solution has been obtained) estimate of the error.

The discretisation process invariably introduces an error, unless the underlying exact solution has a very elementary analytical form. In Chapter 3 it has been said that in the limit when the control volume shrinks to a point, the exact solution of the mathematical model is obtained if the numerical method is convergent, resulting in zero numerical error. This leads some modellers to employ fine meshes in all parts of the flow domain where some important aspect of the flow might manifest itself. The quality of results is generally judged by the invariance of solutions with further refinement, if it is feasible and one can afford the cost of another calculation. This approach is unacceptable in everyday engineering practice, where often even very coarse meshes, just sufficient to describe the main geometrical properties of the computational domain, already have a very large number of computational points, and any further uniform refinement would lead to a problem whose size would surpass the available computer resources.

Pre-solution refinement strategies are usually based on a subjective judgement ('more computational points in the boundary layer' etc.), and work is needed to make the refinement process more objective, user-independent and, as a consequence, more efficient. The fact that coarse-mesh solutions may be adequate in much of the computational domain, where the flow variables have relatively smooth variations, cannot be exploited in traditional fixed-mesh schemes. Adaptive mesh refinement is a technique that has evolved from an attempt to take advantage of this fact. An adaptive procedure adjusts the location of mesh points, or adds and removes mesh points, using feedback from a previous numerical solution of the problem, concentrating computational points in regions of rapid change in the flow variables (e.g. shock waves, contact surfaces, slip streams or boundary layers), which, if not resolved properly, feed large numerical errors into the solution.

In the past two decades researchers have developed many sophisticated adaptive mesh methods for the solution of ordinary differential equations. Considerable interest has appeared during the last decade in developing

and applying adaptive meshing procedures to the numerical solution of partial differential equations. Significant progress has been made in finite-element adaptive mesh methods (Demkowicz et al. [1985], Oden et al. [1986], Oden et al. [1989]). The progress has been slower in the finite-difference (Denny & Landis [1972], Hawken et al. [1991], Altas & Stephenson [1991]) and finite-volume areas (Caruso [1985], Smith [1990]).

The detection of an error at a particular location does not necessarily mean that the source of that error coincides with that location. It might simply be that the error was convected or diffused from elsewhere. So, apart from detecting the error locally, it is necessary to locate its source. If one could be successful in estimating the error magnitude with high accuracy, any further calculation would be redundant. Oden et al. [1989] remarked that, ideally, in deriving error estimates for elliptic problems, one strives to construct estimates such that the following conditions hold:

- the error estimations are computable from given data and an existing finite-volume/element/difference solution.
- the error estimates behave like the error in some suitable norm, in the sense that the constants C_1 and C_2 exist such that

$$C_1 \|\tilde{e}\| \leq \|e\| \leq C_2 \|\tilde{e}\| \quad (5.2)$$

where \tilde{e} is the error estimate and e is the actual numerical error. Bounds such as (5.2) are viewed as desirable in an effective adaptive process. They ensure that the actual error and the error indicator decrease (or increase) at the same rate as the mesh is refined or as the order of approximation is altered. However, it is usually possible to establish bounds such as (5.2) only asymptotically (Oden et al. [1989]); for example for sufficiently small grid spacing ($h \rightarrow 0$) and a large order of approximation ($p \rightarrow \infty$). Also, even though such estimates may hold, the quality of the error estimate may be poor. This quality is measured by the effectivity index

$$\zeta = \frac{\|\tilde{e}\|}{\|e\|} \quad (5.3)$$

which, one hopes, is close to unity and approaches unity as the adaptive process progresses.

In this chapter, an adaptive mesh refinement procedure which modifies a small portion of the mesh without disturbing a large number of adjacent control volumes is presented. In order to apply local refinement, it is necessary to identify regions of the computational grid in which numerical error is unacceptable, and to do this, it is helpful to examine numerical accuracy in more detail. This is done in Section 5.2. In Section 5.3 techniques for improving numerical accuracy are briefly reviewed, and in Section 5.4 some local error indicators are presented. Section 5.5 briefly describes a strategy that uses multigrid and adaptive grid refinement techniques. Finally, Section 5.6 summarises the discussions and analyses in this chapter.

5.2 NUMERICAL ACCURACY

Before examining techniques for improving numerical accuracy, it is useful to examine the cause of the numerical error e . As explained in Chapter 3, the fact that profile assumptions are necessary in order to derive the discretised equations means that the discretisation process inevitably introduces an error unless the underlying exact solution has a profile which is exactly represented by the assumed one. Thus, the origin of numerical error is the fact that the profile assumptions, or differencing schemes, do not represent the true space and time variation of the dependent variable Φ . In the present study time-independent problems are considered, so that the error introduced in time discretisation is not discussed.

The error introduced by discretisation may be analysed by examining the *truncation error*, or magnitude of neglected terms, of a local Taylor series expansion of Φ (as shown by Lai [1982] and Gosman & Lai [1982] for example) if the grid size is sufficiently small. The Taylor series expansion

of an analytic function $\Phi(\underline{x})$ throughout the neighbourhood of the point \underline{x}_0 (Korn & Korn [1968]) can be written as follows:

$$\begin{aligned}\Phi(\underline{x}_0 + \underline{h}) &= \Phi(\underline{x}_0) + \frac{1}{1!} (\underline{h} \cdot \nabla)^1 \Phi(\underline{x}_0) + \frac{1}{2!} (\underline{h} \cdot \nabla)^2 \Phi(\underline{x}_0) + \dots \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} (\underline{h} \cdot \nabla)^n \Phi(\underline{x}_0)\end{aligned}\quad (5.4)$$

This expression is often used for the evaluation of the order of approximation as well as for the estimation of the numerical error.

One may apply Taylor's theorem to the discrete representation of an integral or differential equation in order to recover the original governing integral or differential equation. The higher-order terms in the Taylor series, which were not present in the original governing equation, represent *the truncation* in the numerical solution. The computational nodes should be spaced close enough for these extra terms to be considered as negligible in comparison to the terms of the original equation, otherwise one would obtain the numerical solution of the governing equation with extra terms in it (Hawken et al. [1991]). If odd-order derivatives predominate in the truncation error, the extra terms can cause the new governing equation to have an oscillatory behaviour (Gibbs' oscillations) in the vicinity of sharp gradients of the dependent variable, which the original governing equation does not possess. This effect is known as *numerical dispersion*. If even-order derivatives predominate in the truncation error, smearing of sharp gradients (known as *numerical diffusion*) can result.

Numerical dispersion is a particularly 'visible' problem in flows where diffusive transport is negligible in comparison to convective transport, since there is no dissipative mechanism that would suppress the oscillations generated by the differencing scheme. The usual remedy is to introduce a certain amount of artificial diffusivity, either in an explicit manner or by using a first-order accurate differencing scheme (for example the upwind differencing scheme described in Section 3.4), which is

characterised by the presence of an anisotropic numerical diffusion. There are many differencing schemes reflecting different attempts to select the way and amount of artificial diffusion to be introduced in order to sufficiently suppress these oscillations, and yet not change the physical nature of the original governing equations (e.g. Lax & Wendroff [1964], MacCormack [1969], Van Leer [1982], Harten [1984]). In the present work a blending of upwind and central differencing, as described in Section 3.4 is used to suppress the non-physical oscillations.

The *numerical diffusion* present in the solution of discretised system of governing equations of fluid flow (Patankar [1980], Leschziner [1980]) is another serious manifestation of numerical truncation error and originates in modelling convective transport, as a consequence of neglecting derivatives of first order and higher in the Taylor series expansion (e.g. as in the first-order upwind differencing scheme). An approximation of convective flux through the face j (Section 3.3) is given by:

$$C_j = (\underline{v}_j \cdot \underline{S}_j) (\phi_{P_x} + \nabla \phi_{P_x} \cdot (\underline{x}_j - \underline{x}_{P_x}) + \sum_{n=2}^{\infty} \frac{1}{n!} ((\underline{x}_j - \underline{x}_{P_x}) \cdot \nabla)^n \phi(\underline{x}_{P_x})) \quad (5.5)$$

where P_x is a suitably-chosen point in the computational domain upwind of the cell face j through which the flux is to be calculated.

If one decides to use the upwind-differencing scheme because of the reasons mentioned in Section 3.4, the convective flux (5.5) will be under- or over-estimated by an amount of numerical diffusion that can be approximated by the leading term of the truncation error:

$$C_j^n = (\underline{v}_j \cdot \underline{S}_j) (\nabla \phi_{P_x} \cdot (\underline{x}_j - \underline{x}_{P_x})) \quad (5.6)$$

Looking at expression (5.6) it is easy to see three ways to reduce numerical diffusion. The first is to have the fluid velocity vector \underline{v}_j orthogonal to the cell face vector \underline{S}_j . The second is to chose location P_x such that the gradient vector $\nabla \phi_{P_x}$ is orthogonal to the distance vector $(\underline{x}_j - \underline{x}_{P_x})$ which connects that location with the cell face centre, and the last one is to make this distance vector small. For subsonic flows, the component of the gradient of the dependent variable is usually small in the flow direction.

These facts are often used in reducing numerical diffusion (Syed et al. [1985]), either by creating numerical meshes that are aligned with the flow streamlines (Demirdzic et al. [1980]) or creating 'skew' upwind differencing schemes (Raithby [1976]), for which the location P_x is chosen to be upstream in the direction of fluid velocity vector at the considered cell face. In transonic and supersonic flow simulations, where the large gradients occur in the direction of fluid flow, neither of these strategies can be used, and a combination of local grid refinement and higher-order differencing schemes is the main way of avoiding excessive numerical diffusion.

Although in principle the truncation error exactly defines the numerical error, it cannot easily be evaluated quantitatively. Instead, only the order of accuracy, being the exponent of the mesh size of the leading neglected term, is readily available. While this gives an indication of how quickly the error tends to zero as the mesh is refined, it does not necessarily reflect the magnitude of numerical error, which depends on the magnitudes (rather than orders) of all the neglected terms (and not just the leading one). This explains why improved accuracy is not necessarily guaranteed simply by employing a higher order differencing scheme (Fletcher [1991]).

5.3 METHODS FOR IMPROVING NUMERICAL ACCURACY

Numerical accuracy can be improved by reducing the error in the profile assumptions, and methods for achieving this are usually based on the following approaches:

- 'improving' the accuracy of the discretised equation by using a higher order differencing scheme (p -refinement).
- altering the computational grid so that mesh spacing is made smaller, making the details of the profile assumptions less important (r - and h -refinement).

For a given required solution accuracy it can be more economical to solve a higher-order difference scheme on a coarse grid than a low-order scheme on a finer grid, if the exact 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 severe gradients (Fletcher [1991]) or if the numerical grid is too coarse. An obvious and indisputable way for improving the numerical accuracy is therefore the second approach, so that the distance over which the profile assumptions apply is made smaller, and the details of the assumptions become less important.

Uniform grid refinement, for which mesh spacing is everywhere reduced by subdividing the entire existing mesh, results in a reduction of numerical error at the price of a large computer memory and time increase. A much more efficient approach for improving numerical accuracy is that of *adaptive local grid refinement*, for which grid points are concentrated only in regions of large numerical error, so that the highest grid densities occur in regions in which the solution varies most rapidly.

An adaptive local grid refinement method must have several ingredients (Thompson [1984]), such as:

- a measure of the error in the discrete values that bears some relation to the truncation error.
- a means of increasing the number of grid points (resolution) as indicated by the measure, so as to reduce solution error.
- a means of reducing resolution as indicated by the measure, so as to reduce computer demands.

Adaptive techniques can be classified into two main categories:

- *global* or *mesh movement* methods, in which the total number of grid point is fixed, and the spacing is adjusted to minimise the error (r -refinement). These methods are called global since the entire computational domain is involved in the adjustment process.

- *local* methods or mesh enrichment/depletion, in which cells are added or deleted locally, until the error in the whole solution domain falls below a prescribed minimum (h -refinement).

The problem with mesh movement is that, for a given number of grid points, there is no guarantee of obtaining a sufficiently accurate solution. Thus, mesh enrichment is the preferred approach (Löhner et al. [1985]). Since multigrid methods already make use of a series of progressively fine grids, it is particularly convenient to incorporate local grid refinement simply by confining finer grids to local patches (Smith[1990]) rather than allowing them to cover the whole domain.

5.4 LOCAL REFINEMENT INDICATORS

Before local grid refinement can be employed, it is necessary to identify regions of the computational domain in which the density of the currently-employed mesh is not sufficient to resolve the complexity of flow field, feeding unacceptably-large numerical errors into the solution. This identification is the job of local refinement indicators. As mentioned in the introduction, error estimation is necessary to decide if further refinement is necessary, but may be unreliable indicator for the location and degree of refinement. In particular, the mesh should not be necessarily refined in the regions where the numerical error is large, but in the regions where the source of that error is located.

The governing conservation equation is approximated on a grid with the spacing h by the discrete equation:

$$\mathbf{A}\phi = \mathbf{b} \quad (5.7)$$

where \mathbf{A} is the discrete operator or coefficient matrix. Substitution of (5.1) into (5.7) leads to the error equation:

$$\mathbf{A}\mathbf{e} = \boldsymbol{\tau} \quad (5.8)$$

where τ is the *truncation error source*, given by:

$$\boxed{\tau = A\Phi - b} \quad (5.9)$$

Thus, the numerical error distribution, given by the solution of equation (5.8), is generated by the truncation error source defined by equation (5.9), or *tau error* as it is called by Phillips [1984], which may be used as a local refinement indicator (Berger [1982], Berger & Jameson [1984], Caruso [1985]).

In Section 5.1 the transport nature of the solution error has been mentioned. Equation (5.8) describes that transport, and as one can see its transport properties are defined by the coefficient matrix A of the dependant variable Φ . This equation can be used to calculate the solution error once the distribution of the tau error is known or to identify the regions that are polluted by the error introduced at a certain point of the computational domain. The latter might be useful to someone who is not concerned with the presence of a relatively large error at some location in the computational domain, as long as it does not affect the solution in the important regions.

Although the solution error ϵ is the measure of the quality of the numerical results, its direct use as a local grid refinement indicator would lead to unnecessary fine meshes in regions where the error is large, but its source is small. This is possible because of the transport nature of the solution error, expressed by equation (5.8). Because of this, the distribution of tau error is required information for the efficient detection of the regions where local grid refinement should be employed.

Expression (5.9) presents the tau error as a source term, which should be added to the discretised form of the governing equation in order to get the exact solution. It is useful to recognise that the underlined term of coarse grid FAS equations (4.35) is an approximation of the tau error, which added to the source term of the standard coarse grid equation gives it fine grid accuracy as the solution proceeds.

It should be noticed that in equation (5.8) both ϵ and τ are unknown, since, in practice, the exact value Φ is not available. However,

approximations to the e or τ may be obtained, using an improved solution. This improved solution may be obtained by producing a solution on a finer grid or improving the solution on the currently finest grid using some form of 'postprocessing' (e.g. Richardson extrapolation (Fletcher [1991]) or a higher-order differencing scheme (McGuirk et al. [1980])).

In what follows two techniques for error estimation are presented, one whose variants are widely used in the finite volume community (Caruso [1985], Becker et al. [1988], Thompson & Ferziger [1989], Smith [1990]) and which is based on the comparisons of solutions for different space discretisation, and a novel one based on an estimation of the leading terms of the truncation error and the use of different equation discretisations.

5.4.1 Error Estimation Based on Richardson Extrapolation

For sufficiently fine grids the solution error reduces as the truncation error. This feature motivates Richardson extrapolation, which is used to improve the solution accuracy on a given grid. The basic idea is to add together suitably weighted solutions on successively refined grids to cancel the leading term in the truncation error. The form of the truncation error arises from Taylor series expansion and therefore it is not surprising to find that Richardson extrapolation often requires a relatively fine grid to be effective (Fletcher [1991]).

The spatial variation of the exact solution Φ on two grids with spacing h_1 and h_2 is given by:

$$\Phi(x) = \phi(x, h_1) + h_1^p F(x) + O(h_1^q) \quad (5.10.a)$$

$$\Phi(x) = \phi(x, h_2) + h_2^p F(x) + O(h_2^q) \quad (5.10.b)$$

where $\phi(x, h_i)$ is the approximate solution on grid with spacing h_i , p is power of the leading h_i term in the expression for the truncation error, and $q \geq p+1$. Subtracting expressions (5.10.a) and (5.10.b) it is possible to obtain an approximation for $F(x)$:

$$F(x) \equiv \frac{\phi(x, h_2) - \phi(x, h_1)}{h_1^p - h_2^p} \quad (5.11)$$

which, after substitution into equation (5.10.a), gives an improved estimate for ϕ :

$$\phi(x,0) = \phi(x,h_2) \frac{\left(\frac{h_1}{h_2}\right)^p}{\left(\frac{h_1}{h_2}\right)^p - 1} - \phi(x,h_1) \frac{1}{\left(\frac{h_1}{h_2}\right)^p - 1} \quad (5.12)$$

which is q order accurate. An expression for the error approximation follows from equation (5.12):

$$e(x,0) = \phi(x,0) - \phi(x,h_2) = \frac{\phi(x,h_2) - \phi(x,h_1)}{\left(\frac{h_1}{h_2}\right)^p - 1} \quad (5.13)$$

This approach can be applied to estimate the solution error for each dependent variable. However, for fine and coarse grid arrangement adopted in the present study (Section 4.6.1) the locations of the computational points on the coarse and fine grid do not coincide and interpolation to fine grid locations is required. In the present study the prolongation operator (4.43) defined in Section 4.6.3. is used to transfer the coarse grid solution to the fine grid.

The improved solution $\phi(x,0)$ obtained by (5.12) can be used to estimate the truncation error (5.9) for the governing equations, and the numerical error approximation defined by (5.13) can be monitored as a measure of numerical error present in the solution.

5.4.2 Significance of Pressure-Coupling Term in the Expression for Cell Face Velocity as an Error Estimate for Pressure

In Section 3.6.1 it is shown that pressure-velocity decoupling on a collocated grid may be avoided by introducing a pressure-coupling term, which vanishes in the limit of a fine computational grid or if the pressure gradient varies linearly in space. However, in general this term has a finite value and at this point it will be further discussed and its significance as an error indicator in the pressure calculation will be presented. To the author's knowledge this is novel approach.

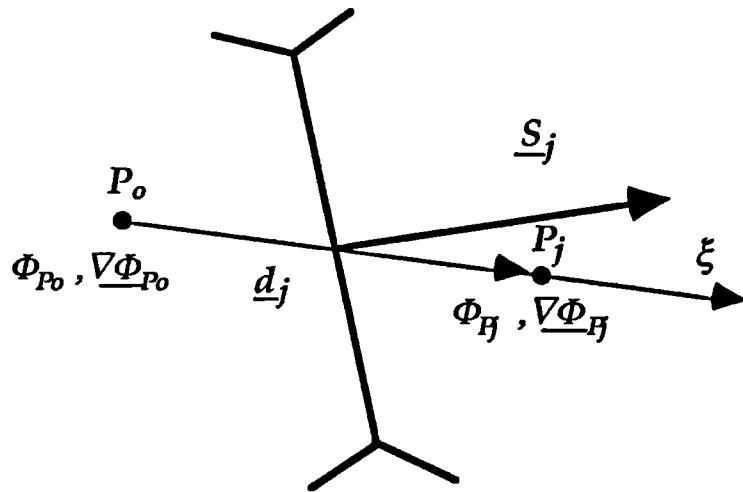


Figure 5.1 One-dimensional truncation error analysis for variable Φ
($\Phi = v_i, p, T, k, \varepsilon, \dots$)

A one-dimensional Taylor series analysis performed in direction ξ defined by the relative position vector d_j connecting two computational points that have in common cell face j (Figure 5.1) leads to the following expressions:

$$p_{P_o} = p(\xi) - \frac{\partial p}{\partial \xi} \Delta \xi + \frac{\partial^2 p}{\partial \xi^2} \frac{\Delta \xi^2}{2!} - \frac{\partial^3 p}{\partial \xi^3} \frac{\Delta \xi^3}{3!} + O(\Delta \xi^4) \mid \xi = \Delta \xi \quad (5.14.a)$$

$$p_{P_j} = p(\xi) + \frac{\partial p}{\partial \xi} \Delta \xi + \frac{\partial^2 p}{\partial \xi^2} \frac{\Delta \xi^2}{2!} + \frac{\partial^3 p}{\partial \xi^3} \frac{\Delta \xi^3}{3!} + O(\Delta \xi^4) \mid \xi = \Delta \xi \quad (5.14.b)$$

$$\left(\frac{\partial p}{\partial \xi} \right)_{P_o} = \frac{\partial p}{\partial \xi} - \frac{\partial^2 p}{\partial \xi^2} \Delta \xi + \frac{\partial^3 p}{\partial \xi^3} \frac{\Delta \xi^2}{2!} + O(\Delta \xi^3) \mid \xi = \Delta \xi \quad (5.14.c)$$

$$\left(\frac{\partial p}{\partial \xi} \right)_{P_j} = \frac{\partial p}{\partial \xi} + \frac{\partial^2 p}{\partial \xi^2} \Delta \xi + \frac{\partial^3 p}{\partial \xi^3} \frac{\Delta \xi^2}{2!} + O(\Delta \xi^3) \mid \xi = \Delta \xi \quad (5.14.d)$$

where $\Delta \xi = \frac{|d_j|}{2}$ and $\frac{\partial p}{\partial \xi} = \nabla p \cdot \frac{d_j}{|d_j|}$.

Subtracting (5.14.a) from (5.14.b) and (5.14.c) from (5.14.d) leads to the expressions:

$$\frac{\partial^3 p}{\partial \xi^3} \frac{\Delta \xi^2}{3!} = \frac{p_{Pj} - p_{Po}}{|\underline{d}_j|} - \frac{\partial p}{\partial \xi} + O(\Delta \xi^3) \quad (5.15.a)$$

$$\frac{\partial p}{\partial \xi} = \frac{1}{2} \left(\left(\frac{\partial p}{\partial \xi} \right)_{Po} + \left(\frac{\partial p}{\partial \xi} \right)_{Pj} \right) - \frac{\partial^3 p}{\partial \xi^3} \frac{\Delta \xi^2}{2!} + O(\Delta \xi^3) \quad (5.15.b)$$

Substituting (5.15.b) into (5.15.a) one obtains the following expression for the truncation error of a second order approximation for the pressure variation:

$$\frac{\partial^3 p}{\partial \xi^3} \frac{|\underline{d}_j|^2}{12} + O(\Delta \xi^3) = \left(\frac{1}{2} ((\nabla p)_{Po} + (\nabla p)_{Pj}) \cdot \frac{\underline{d}_j}{|\underline{d}_j|} - \frac{p_{Pj} - p_{Po}}{|\underline{d}_j|} \right) \quad (5.16)$$

It is easy to see that the expression on the right hand side of equation (5.16) is identical to the pressure-coupling term appearing in the expression for the cell-face velocity (3.29). Once the solution of the discretised governing equations is achieved, approximations of pressure and its gradient vector are readily available, and these can be used in (5.16) to give an estimate of the truncation error introduced by the second order discretisation of pressure term in momentum equation.

5.4.3 A New Error Estimation Technique Based on Higher Order Derivatives

The error estimation technique described in Section 5.4.1 needs solutions on two computational grids of different resolution. This is not an important issue in an 'academic' environment, where the size and complexity of the considered fluid flow domains need not require a large number of computational points. However, industrial CFD problems often need very large numerical meshes to describe the major geometrical features. In this situation, an attempt to increase mesh resolution uniformly throughout the computational domain, in order to assess the numerical error, may easily generate a mesh that is beyond the available

computer resources. It would therefore be very useful to have a local error estimate obtained from the available single grid calculation, so that the results may be assessed and local refinement performed in regions where necessary, until the error is reduced below prescribed level or the available computer resources are exhausted.

The new error detection technique described below enables *a posteriori* local error estimation, extracting the information about the spatial variation of the dependant variables from the solution on a given grid. This provides an error estimate even on the coarsest grid. Thus, even in a transient calculation at every time step this measure enables refinement of the mesh in regions where large variable variations are detected, and coarsening when these large gradients have died out. This new error-detection technique is inspired by the analysis of the pressure-coupling term described in the previous section.

As already noted, the convective and diffusive fluxes of momentum, energy or any other conserved property passing through the cell face may be over- or under-estimated in some regions as a result of an inaccurate assumption about the spatial variation of the dependent variables used in evaluating these fluxes. The values of these over- or under-estimations are proportional to the truncation error (Section 5.2). If the variation of the dependant 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 truncation error and that its leading term will be the dominant contributor to the magnitude of the truncation error and this leading term is properly evaluated. A final condition on the validity of our assumption (that if the truncation error is small, the solution error is small) is that the equations must be well-posed in the sense that a small change in a term will have only a small effect in the determination of the solution. Whilst this seems a plausible assumption to make, the author knows of no formal proof for the type of governing equations under consideration.

McGuirk and Rodi [1978] and McGuirk et al. [1980] have shown how the regions in the flow with large numerical diffusion can be identified *a posteriori* by evaluating the amount of local numerical diffusion present

in a solution obtained with a given grid using an estimate of the magnitude of the local truncation error, using the difference between upwind and central difference expressions for the convection terms. The use of this error estimation is limited to convection dominated problems (error resulting in modelling diffusion is not considered), when the convection is modelled with the first-order accurate upwind differencing scheme.

In the analysis of the pressure-coupling term presented in the previous section, an estimate of the truncation error proportional to the third derivative of pressure, in a direction connecting two computational points, is evaluated using the values and the gradients of pressure at these points. This is very convenient, since the evaluation of gradient described in Section 3.4 does not depend on the cell topology, and once it is available, this analysis becomes simple and compact. In what follows an analysis is presented that leads to conclusions similar to those of the pressure-coupling term (5.16) analysis, but its formulation is more general and it can be used to estimate the error resulting from modelling the convective and diffusive fluxes of a conserved variable for a first- and second-order discretisation scheme.

Again, the values of variable Φ and its gradient at two neighbour locations are used in analysis and instead of looking at the distribution of Φ in space, the variation only in direction ξ , which connects the neighbouring points is considered (Figure 5.1). Projecting the gradient of Φ onto the direction 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.

$$\phi(\xi) = a_0 + a_1 \xi + a_2 \xi^2 + a_3 \xi^3 \quad (5.17)$$

The coefficients a_i are to be determined in such a way that (5.17) satisfies the following conditions:

$$\begin{aligned} \xi = 0: \quad & \Phi = \Phi_{P_0} \quad ; \quad \xi = |d_j|: \quad \Phi = \Phi_{P_j} \\ \xi = 0: \quad & \nabla \Phi^\xi = (\nabla \Phi)_{P_0}^\xi \quad ; \quad \xi = |d_j|: \quad \nabla \Phi^\xi = (\nabla \Phi)_{P_j}^\xi \end{aligned} \quad (5.18)$$

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

$$(\nabla\Phi)_{Pj}^\xi = (\nabla\Phi)_{Pj} \cdot \underline{d}_j \quad (5.19)$$

The coefficients a_i that satisfy conditions (5.18) are given by:

$a_0 = \Phi_{P_0}$ $a_1 = (\nabla\Phi)_{P_0}^\xi$ $a_2 = 3 \frac{\Phi_{Pj} - \Phi_{P_0}}{ \underline{d}_j ^2} - \frac{(\nabla\Phi)_{Pj}^\xi + 2(\nabla\Phi)_{P_0}^\xi}{ \underline{d}_j }$ $a_3 = -2 \frac{\Phi_{Pj} - \Phi_{P_0}}{ \underline{d}_j ^3} + \frac{(\nabla\Phi)_{Pj}^\xi + (\nabla\Phi)_{P_0}^\xi}{ \underline{d}_j ^2}$	(5.20)
---	--------

As one would expect, if ϕ is replaced by p , the expression for the third derivative of (5.17), is identical to the leading term in truncation error (5.16), which is result of a second order approximation of pressure gradient. Expression (5.16) can be used for estimation of the dependent variable and its gradient in direction \underline{d}_j at the cell face j :

$$\tilde{\phi}_j = a_0 + a_1 \xi_j + a_2 \xi_j^2 + a_3 \xi_j^3 \quad (5.21)$$

$$\tilde{\nabla}\phi_j \cdot \frac{\underline{d}_j}{|\underline{d}_j|} = a_1 + 2a_2 \xi_j + 3a_3 \xi_j^2$$

where ξ_j is the coordinate that defines the intersection of the distance vector \underline{d}_j and the cell face plane.

Now it is possible to evaluate mass, convection and diffusion fluxes of a conserved property based on (5.21) as follows:

$$\tilde{F}_j = \tilde{\rho}_j \tilde{\underline{v}}_j \cdot \underline{S}_j \quad (5.22)$$

$$\tilde{C}_j = \tilde{F}_j \tilde{\phi}_j \quad (5.23)$$

$$\tilde{D}_j = -\tilde{\lambda}_{\phi_j} \left(\tilde{\nabla}\phi_j \cdot \frac{\underline{d}_j}{|\underline{d}_j|} \right) \frac{\underline{d}_j}{|\underline{d}_j|} \cdot \underline{S}_j \quad (5.24)$$

where $\tilde{\rho}_j$, \tilde{v}_j and $\tilde{\lambda}_{\Phi j}$ are density, velocity and diffusivity at the cell face j obtained using (5.21).

In general, these fluxes are different from those obtained during the calculation, since the profile (5.17) is different from the one used during the discretisation described in Section 3.4. Summing the differences between these fluxes over the surfaces bounding the control volume leads to an approximation of the tau error at P_o :

$$\tilde{\tau}_{P_o} = \sum_{j=1}^n ((\tilde{C}_j - C_j) + (\tilde{D}_j - D_j)) \quad (5.25)$$

The source $\tilde{\tau}_{P_o}$ approximated in this way will be large in the regions where the variation of dependent variable is large, and zero where the dependent variable does not change in space or varies linearly.

An approximation of the solution error as defined by equation (5.1) is obtained by dividing tau by the central coefficient a_o of coefficient matrix A (equation 5.8). The approximation arises because the surrounding values of solution error e_{P_j} are also modified by the tau error, but this explicit update of e_{P_o} does not take into account changes in its neighbourhood (Smith [1990]). So an approximation for the solution error at point P_o based on the approximation of the tau error (5.25) is given by:

$$\tilde{e}_{P_o} = \frac{\tilde{\tau}_{P_o}}{a_o} \quad (5.26)$$

An estimation of the error introduced by approximating the value of dependent variable at the cell face j is given by:

$$\tilde{e}_j = \tilde{\phi}_j - \phi_j \quad (5.27)$$

A consequence of the above one-dimensional analysis is that the error in modelling cross-diffusion (Section 3.5) cannot be estimated. However, the error detected in modelling normal diffusion should provide

sufficient information about where further local grid refinement is necessary. Visiting all faces of a considered cell gives an estimate of the error distribution in several independent directions, so that the procedure described does not lose information about the variation in space of the variable and its error. A three-dimensional study, similar to the one-dimensional analysis described above, would introduce additional complexity, without providing useful additional information for problems dominated by convection or where severe cross-diffusion is not present.

The error-detection technique, presented in this section, does not take into account interequation coupling and the non-linear nature of the governing equations. It is important to realise that inaccuracy in resolving the spatial variation of one dependent variable will cause error in fields of all others. This aspect is relevant for estimation of the absolute solution error, which would be obtained by comparing the current solution with the grid-independent one. However, an estimation of the truncation error provides a measure of how well each of these fields is represented on a given mesh and where this representation is not satisfactory. When all fields are sufficiently resolved, one hopes that the maximum estimated solution error will be of the same order as the exact one based on the grid independent solution.

The error (5.27) has additional information about direction, and is valuable if, instead of refining a control volume in all directions, one decides to refine only in those directions where the variation of dependant variable is large (Figure 5.2). The whole procedure might reduce the total number of control volumes in 'parabolic' flows, where the large variations of dependent variables occur in the cross-flow direction only. In the present study cells were 'isotropically' refined wherever the error estimation (5.26) was larger than the prescribed value, since directional splitting of control volumes is an additional overhead on a mesh generator, demanding development of particular strategies for different cell topologies and situations, as well as additional book-keeping for the information exchange between the coarse and fine grid necessary for a multigrid method.

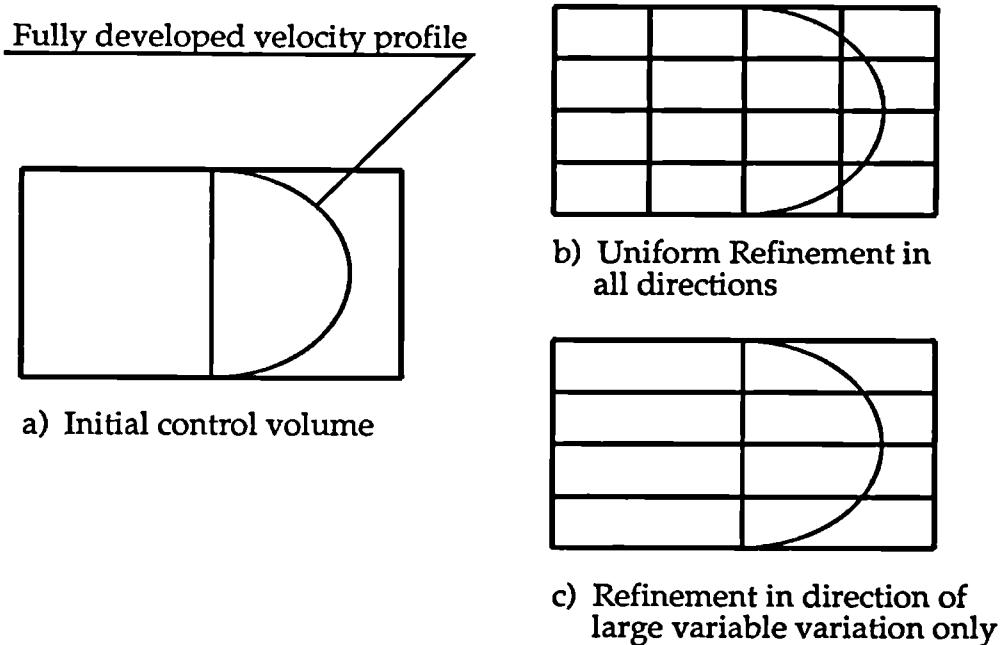


Figure 5.2 Consequences of uniform cell refinement and refinement in direction of large variable variation only.

5.5 NORMALISATION OF THE ERROR INDICATOR

A question which has not been answered so far is at what level of truncation error should the grid be refined locally, or what level may be safely neglected, in order to achieve the desired accuracy?. Obviously the actual value will depend on the normalisation of the truncation error. In the present study, the normalised value of the truncation error defined by equation (5.9) or (5.25) is obtained as follows:

$$\tilde{\tau}_\phi^* = \frac{\tilde{\tau}_\phi}{a_0 \phi_{ref}} 100 \text{ [%]} \quad (5.28)$$

where a_0 is the central coefficient, given by equation (3.21) and ϕ_{ref} is a reference value. In this way the resulting normalised tau error is an approximation to the solution error e at this point, in a similar way that \tilde{e}_{P_0} is related to $\tilde{\tau}_{P_0}$ by expression (5.26). ϕ_{ref} might be equal to the value of

dependant variable at the location where the tau error is calculated; or it can be a typical value in the computational domain or a subdomain. If the local value of dependant variable happens to be small, a large normalised tau error might be produced, even if the difference between the computed and exact solutions is insignificant compared to the average or maximum value ϕ . So normalisation based on the local value of the dependant variable can be misleading. In the present study the normalisation was based on the maximum value of the dependant variable in the considered computational domain.

Specifying the refinement criterion T^* (usually 0.5% - 5%) such that the relationship:

$$\tau < T^* \quad (5.29)$$

is satisfied at all computational points is equivalent to the demanding that the maximum approximated solution error is smaller than T^* when the final mesh is constructed.

Since pressure is not a conserved property, strictly speaking there is no term equivalent to the tau error, which could be used for the error estimation in the pressure calculation. Thus in the present study expression (5.16) is used for the error estimation of pressure, but combined, in a novel manner, with the cell-face velocity equation (3.29), which was used to express this error in terms of velocity: this was very useful, since the governing equations for incompressible flow do not fix the pressure absolutely but only determine it to within an arbitrary constant and therefore solution error estimates for pressure cannot be normalised easily (Thompson & Ferziger [1989]). Once the error is expressed in terms of velocity, it is easy to relate it to the reference velocity value \underline{v}_{ref} i.e.

$$e_p^* = 100 \cdot \frac{K \left(\frac{1}{2} ((\nabla p)_{P_o} + (\nabla p)_{P_j}) \cdot \frac{\underline{d}_j}{|\underline{d}_j|} - \frac{p_{P_j} - p_{P_o}}{|\underline{d}_j|} \right)}{|\underline{v}_{ref}|} [\%] \quad (5.30)$$

where K is defined by expressions (3.32). The error estimate (5.30) is calculated for each cell face, and the judgement on further refinement of a particular cell is based on the maximum detected value of e_p^* and refinement criterion T^* .

5.6 OVERALL ALGORITHM FOR MULTIGRID AND LOCAL GRID REFINEMENT

All the necessary individual components have now been devised for an efficient adaptive discretisation and solution algorithm. Now these components will be assembled.

- An initial mesh must be produced. This mesh may be just sufficient to describe the major geometrical features of the considered computational domain. If the surface description is given in some way, then everything that follows can be done automatically.
- Once the initial mesh is in place, the discretised versions of the governing equations of fluid flow are solved in a sequential manner (Section 3.8).
- When a certain level of convergence is reached, the solution is examined. The cells at which the normalised truncation error estimate (Sections 5.5) 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 the safety margin d , together with T^* , determines the part of the domain covered by the refined grid components. The safety distance d is related to the characteristic cell size h of the currently finest grid (typically 2-4 h). The cells that are marked are refined in the manner described in Section 4.6.1.
- At this stage two numerical grids of different resolution are available, covering the same computational space. The fields of dependent variables are extrapolated from the coarse to the fine grid and the V multigrid cycle on those two grids is started as described in Section 4.6.5.

The whole procedure of adaptive refinement of the currently finest mesh, and solution of the corresponding discrete system of algebraic equations using multigrid acceleration is repeated until the error estimate is below a desired value, or available computer resources are depleted.

In an early stage of the algorithm, when the numerical mesh might be very coarse, the error estimate may not be very reliable. In this case the error estimation may be repeated on a finer grid, and if some of refinements are found not to be necessary, they can be reversed.

5.7 CLOSURE

In this chapter, the technique of local grid refinement for efficiently reducing numerical error, and its combination with the multigrid method, has been described. The benefits of unstructured meshing, solution-adaptation and a discretisation practice that is independent of cell topology are combined to produce a very powerful analytical tool.

Firstly in this chapter, numerical accuracy in terms of truncation error and the cause of numerical diffusion was examined. Methods for improving numerical accuracy were discussed, and the advantages of local grid refinement, especially when combined with multigrid solvers, were outlined. In order to apply local grid refinement, it is necessary to identify regions requiring refinement. Hence, refinement indicators were examined, and a novel error estimation technique was presented. According to the error indication, the mesh is modified locally by adding new computational points. The addition of new mesh points into the mesh is straightforward, and is achieved by splitting marked control volumes into a number of smaller control volumes as described in Section 4.6.1. For the deletion of mesh points the algorithm is reversed.

Before further local grid refinement may be performed two questions have to be answered: is further refinement necessary and if it is, which are the regions where local grid refinement should take place? In order to answer the first question one should know the distribution or the maximum value of the solution error ε . In order to answer the second question one should know the distributions of the solution error source or the truncation error τ . Although truncation and solution error are closely

linked, these two have different natures. The solution error may be regarded as a transported variable, which is generated by the truncation error (Section 5.4). The governing equations of CFD are non-linear and coupled, and each of these equations has its corresponding error equation. These error equations make a system of coupled non-linear equations, whose coefficients are based on the dependent variable fields. Because of the non-linear nature of those equations, it is very difficult to quantify the influence one on another. Since there is uncertainty about truncation error estimation, it does not appear worthwhile to solve this system, and the estimation of the maximum absolute error obtained from the truncation error estimation, as explained in Section 5.5 should be satisfactory for most purposes. However, absolute error estimation based on Richardson extrapolation (Section 5.4.1) is a method where the solution error comes as a result of comparing the solutions on coarse and finer meshes. Since the solution on the finer mesh advances towards a grid independent solution, the solution error estimation has information about the non-linear nature of the problem which is solved and about the transport properties of the solution error. Because of this, it might be useful to use this technique, when solution on two grids of different resolution are available, since it is probably more reliable for estimating absolute solution error.

As the adaptive mesh refinement progresses, a hierarchy of numerical grids is constructed. The interaction and information exchange between these grids, as described in Chapter 4, forms a multigrid algorithm, the performance of which is demonstrated in the next chapter.

Chapter 6

Case Studies

6.1 INTRODUCTION

In the previous three chapters numerical techniques for the solution of problems relevant to fluid dynamics have been examined, using multigrid to accelerate the convergence of iterative solvers, and local mesh refinement to economically resolve regions where large sources of numerical error occur. Along the way, choices have been made on differencing schemes, treatment of pressure-velocity coupling and variable arrangement, as well as coarse and fine grid structure, restriction and prolongation operators, relaxation schemes and multigrid cycles, for use in this study.

In this chapter the accuracy and efficiency of these techniques will be demonstrated and the properties of some components of the methodology will be investigated. Test cases which have well-established numerical or experimental results are selected. The effectiveness of the multigrid solver will be investigated by performing calculations on meshes of different numerical resolution, and the multigrid convergence rate will be compared with single-grid results. The solution error is also monitored so that the grid can be locally refined until the error estimate is below a desired level, or until the available computer resources are exhausted.

In all cases the error estimation is performed on the velocity and pressure fields. In addition, for the prediction of turbulent flows, estimation is made of the error present in the turbulent kinetic energy and

dissipation fields. The local grid refinement was performed when the estimated error (Sections 5.4) in any of aforementioned fields exceeded the maximum value prescribed, which was 1% for the most of the cases.

Since the high-Re-number $k-\varepsilon$ eddy-viscosity model with wall functions (Sections 2.4 & 3.7.4) was used in the present calculation, the near-wall y^+ was also monitored and refinement was stopped before the near-wall nodes penetrated into the viscous sublayer (i.e. $y^+ < 30$), where the log-law is no longer valid.

Classical multigrid methods were originally developed for elliptic partial differential equations and systems. For such problems these methods have proved to be extremely efficient. When applied to nonelliptic problems (or problems which have a nonelliptic component) such as high-Reynolds number flows, however, performance seems to deteriorate significantly, due in part to the fact that the solutions become more complex (Brandt & Yavneh [1993]).

Slow convergence of multigrid cycles can generally be traced to at least one of two major causes: poor smoothing or poor coarse-grid approximation of the fine-grid problem. Smoothing, which is reducing the amplitude of high-frequency errors that cannot be treated on the next coarser grid (Sections 4.2.4 and 4.3), is dealt with extensively in most publications on multigrid solvers for incompressible flows. There are not so many articles published which are concerned with poor coarse-grid approximations (Brandt & Yavneh [1993]), and none, to the author's knowledge, which address this problem specifically in relation to turbulent flow calculations.

Turbulent kinetic energy and its dissipation rate are two quantities which depend very much on the mesh resolution. Since the field of eddy viscosity depends on these two fields (equation 2.28), the effective transport properties of the medium, could look very different on the coarsest and the finest mesh. Consequently, the representation of the fine-grid problem on a coarse mesh could be poor and the corrections obtained on that mesh interpolated to the fine grid misleading. A remedy, which is used in this study, is to calculate the eddy viscosity on coarse grids

interpolating its solution from the finest grid. Although k and ε are not used to calculate the eddy viscosity on coarse grids, their transport equations are solved and corrections are added to the fine grid solution. If corrections were such that they will cause negative values for k and ε , corrections are set to zero.

The standard way of presenting multigrid convergence results is a work unit (Brandt [1977]). A work unit is the equivalent amount of computational work needed to do one outer iteration on the finest grid using the multigrid smoother. However, in the present work, the total computer (CPU) run time (the information which an engineer is primarily interested in), and number of fine grid iterations are given.

In all examples, a solution is held to have converged when the absolute sum of the residuals of momentum and mass (and turbulent kinetic energy for turbulent cases) fell below 0.001 of corresponding momentum- and mass-flux scales (Section 3.8). To enhance stability, under-relaxation was employed with related factors being 0.8 for velocity, 0.2 for pressure and 0.8 for turbulence parameters. For all calculations (those marked as UD or CD), the convection of k and ε was modelled using the upwind differencing (UD) scheme, to avoid generation of negative values when calculations were performed on the coarser grids. The convergence of the inner iterations was monitored as well, and iteration was stopped once the residual became 0.3 of its initial value.

In the next section the well-known laminar two-dimensional driven cavity flow at Reynolds numbers of 100 and 1000 will be considered. After that, more challenging turbulent flows will become the centre of interest. First, turbulent two-dimensional flow through an orifice plate will be studied in Section 6.3. Three-dimensional turbulent flow over a prismatic obstacle will be discussed in Section 6.4. Finally, the flow inside a gas turbine combustor, which regarding the complexity of geometry and the flow properties is a problem which has features of real-life engineering calculations, will be presented in Section 6.5, followed by closing remarks given in Section 6.6.

6.2 TWO-DIMENSIONAL LID DRIVEN CAVITY FLOW

Laminar driven-cavity flow has long been a standard test problem for Navier-Stokes codes. The problem is characterised by its elliptic and non-linear nature, in common with many engineering flows, and so provides a good test of the computational procedure since there are no uncertainties about the geometry, boundary conditions and turbulence modelling. Since a Cartesian orthogonal mesh is used, it is inevitable that the streamlines are inclined to the mesh in some regions, and that the numerical diffusion becomes a significant source of error.

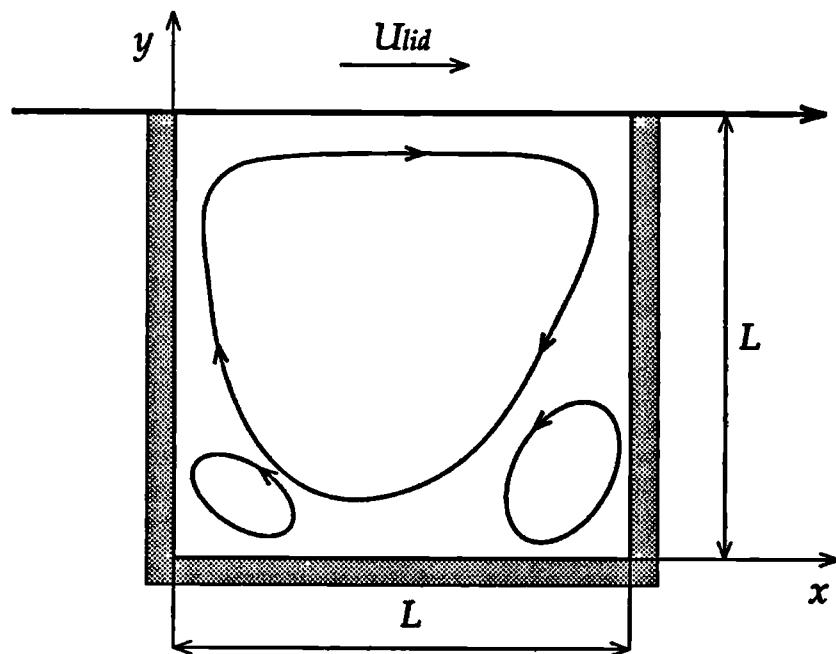


Figure 6.1 Driven cavity test case

The numerical set-up consists of a two-dimensional square cavity with the no-slip boundary condition at the impermeable walls (Figure 6.1). The Reynolds number is based on the width of the cavity and the velocity of the top wall:

$$Re = \frac{\rho L U_{lid}}{\mu} \quad (6.1)$$

This test case has been studied extensively and accurate results have been obtained by various authors (e.g. Ghia et al. [1982], Vanka [1986], Thompson & Ferziger [1989], Smith [1990]) and it is important that the present method shows similar convergence behaviour.

Ghia et al. [1982] used a multigrid technique with a coupled strongly-implicit relaxation scheme and second-order discretisation of the vorticity-stream function formulation of the Navier-Stokes equations. Numerical solutions up to $Re=10,000$ were obtained with 257×257 grid nodes and the fact that the multigrid convergence rate remains almost constant was demonstrated. Vanka [1986] solved the Navier-Stokes equations in primitive variables using upwind differencing, a coupled block-implicit multigrid procedure and symmetrical coupled Gauss-Seidel smoothing technique on a staggered grid and obtained solution times which varied linearly with the number of grid points. A colocated variable arrangement was adopted by Barcus et al. [1987], who applied multigrid, with an uncoupled strongly-implicit relaxation scheme (Stone [1968]), to the SIMPLE algorithm and obtained solutions in 10.3% and 2.1% of the single grid computing time on 40×40 and 80×80 grids, respectively. Thompson and Ferziger [1989] and Smith [1990] presented an automatic adaptive refinement technique coupled to the multigrid approach as an efficient and stable solution strategy for solving the steady-state Navier-Stokes equations, the former on a staggered mesh and latter on a colocated grid. In both cases the refinement criterion was based on the local truncation error as explained in Section 5.4.1. The adaptive refinement approach reduced the computer memory and CPU time by between about 20% and 75% of the requirements of the multigrid method with uniformly-refined grids.

In the present work calculations were performed for Reynolds numbers of 100 and 1000. Apart from the test of multigrid efficiency and performance of different components of the multigrid algorithm, the effects of local grid refinement, based on the methods described in Chapter 5, were tested. In contrast to the work presented in literature so far (i.e. Thompson & Ferziger [1989] and Smith [1990]), the present one does not require explicitly-treated internal boundary conditions at the interface between the locally-refined patches and the rest of the domain, which makes the present calculation more implicit. The effects of mesh non-

orthogonality were not specifically investigated, but the convergence rate for the locally-refined meshes, which were effectively non-orthogonal, was the same as those for the uniformly-refined and consequently orthogonal meshes.

In addition to the single-grid method (SGM) for which calculations starts with zero initial estimates for all variables, a prolongation scheme (PGM) was also employed (Barcus et. al [1987]), where extrapolated coarse grid solutions were used as starting values for the iterations on the fine grid. Since the PGM is a component of the FMG procedure, the results allow one to distinguish the accelerations due to the multigrid method and to the PGM. The influence of prolongation operators described in Section 4.6.3 is tested as well. The multigrid algorithms FMG* and FMG were based on the zero- (4.42) and first-order (4.43) prolongation operators respectively.

6.2.1 Multigrid Results

To assess the accuracy of the present results, it was decided to compare them with those of Ghia et al. [1982], which were demonstrated to agree well with other reported solutions, and are supplied in tabulated, as well as in graphical form. A grid-independence study was performed using uniform grids ranging from 4×4 to 256×256 control volumes (Figure 6.2). These meshes were used for multigrid as well. The residuals in the continuity and momentum equations were normalised with momentum and mass fluxes based on the velocity of the lid U_{lid} and dimension of the cavity L , i.e. with $\rho L^2 U_{lid}$ and $\rho L^2 U_{lid}^2$ respectively.

The streamline patterns for the two-dimensional cavity at Reynolds numbers 100 and 1000 are shown on Figure 6.3. This figure shows the primary and two secondary eddies. Table 6.1 and Figure 6.4 give the values of the stream function at the centre of the primary eddy for the upwind and central difference schemes at the same Reynolds numbers as function of the grid size. These parameters are fairly sensitive indicators of the accuracy of the solution. The results are compared with those obtained by Ghia et al. [1982] who used a second-order vorticity-stream function approach with a grid size of 128×128 . This table shows that the central-

differencing solution converges much more rapidly with grid size than the upwind-differencing solution. It is necessary to use approximately 16 times more grid points with the upwind scheme to achieve the same accuracy as with the central differencing scheme.

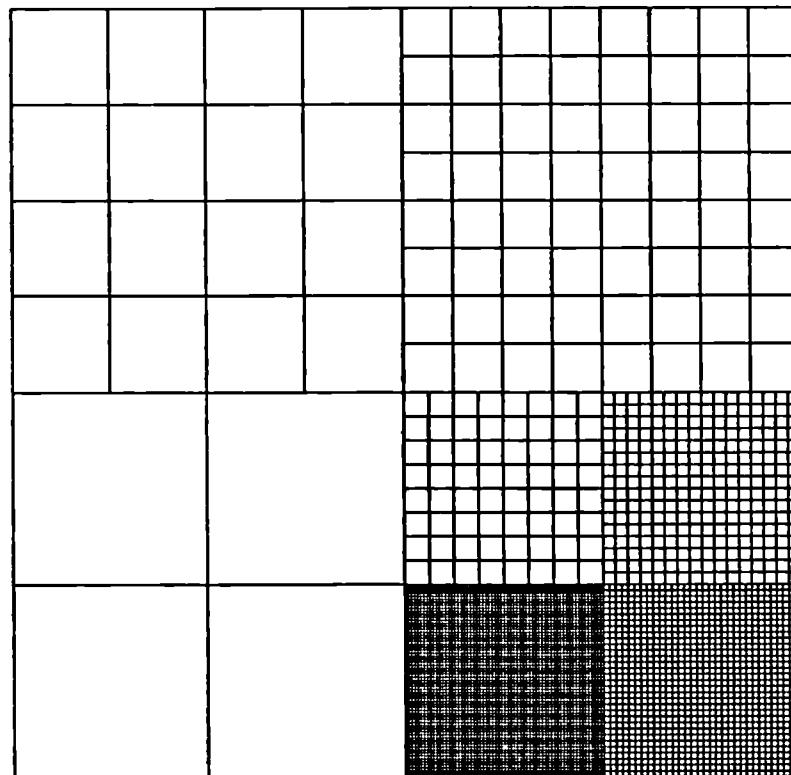


Figure 6.2 Segments of seven numerical grids used in calculation

Similar conclusions follow from the profiles of the u and v velocity vector components along the vertical and horizontal planes passing through the geometric centre of the cavity, which are presented on Figures 6.5 and 6.6. The cavity flow with $Re=1000$ needs a 128×128 numerical mesh and the use of central-differencing to obtain numerically-converged (grid independent) profiles. The upwind differencing solution is quite far away from the numerically-converged one even for the 256×256 mesh. The velocity variation for $Re=100$ is much smoother, and a mesh as coarse as 64×64 and the use of central differencing are sufficient to achieve grid-independent profiles (Figure 6.5).

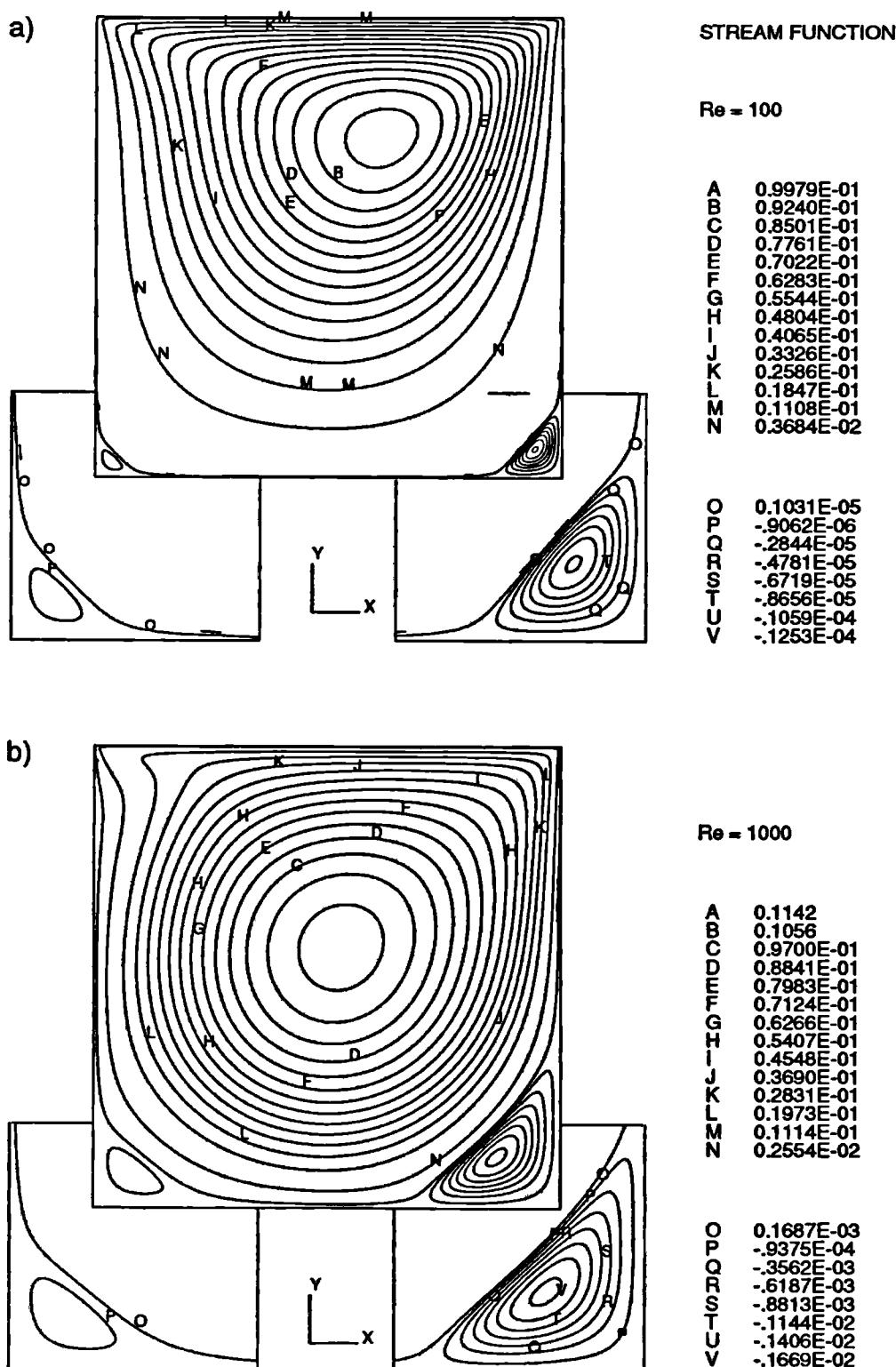


Figure 6.3 Streamlines for the $Re=100$ a) and $Re=1000$ b) cavity flow

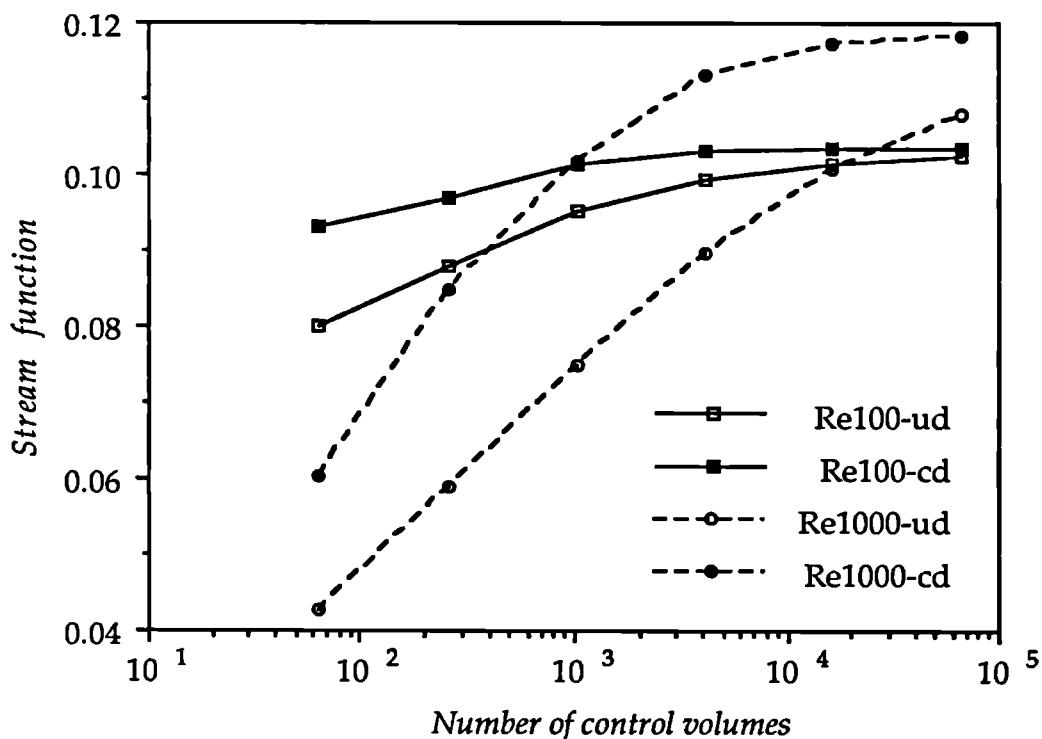


Figure 6.4 Streamfunction at the centre of the primary vortex

Re	Scheme	8x8	16x16	32x32	64x64	128x128	256x256	Ghia
100	UD	sf 0.0800 x 0.625 y 0.750	sf 0.0880 x 0.625 y 0.750	sf 0.0951 x 0.593 y 0.750	sf 0.0993 x 0.609 y 0.750	sf 0.1014 x 0.617 y 0.742	sf 0.1024 x 0.613 y 0.738	sf 0.1034 x 0.617 y 0.734
	CD	sf 0.0930 x 0.626 y 0.750	sf 0.0968 x 0.625 y 0.750	sf 0.1013 x 0.625 y 0.750	sf 0.1030 x 0.609 y 0.734	sf 0.1034 x 0.617 y 0.734	sf 0.1035 x 0.613 y 0.738	
	UD	sf 0.0427 x 0.625 y 0.625	sf 0.0591 x 0.625 y 0.563	sf 0.0748 x 0.563 y 0.594	sf 0.0897 x 0.547 y 0.563	sf 0.1008 x 0.539 y 0.570	sf 0.1080 x 0.535 y 0.566	sf 0.1179 x 0.531 y 0.562
	CD	sf 0.0604 x 0.625 y 0.500	sf 0.0847 x 0.563 y 0.563	sf 0.1016 x 0.511 y 0.563	sf 0.1130 x 0.531 y 0.563	sf 0.1171 x 0.531 y 0.563	sf 0.1183 x 0.531 y 0.566	
sf - stream function; x - x coordinate of vortex centre; y - y coordinate of vortex centre								

Table 6.1 Values of the streamfunction at the centre of the primary vortex and the coordinates of that centre

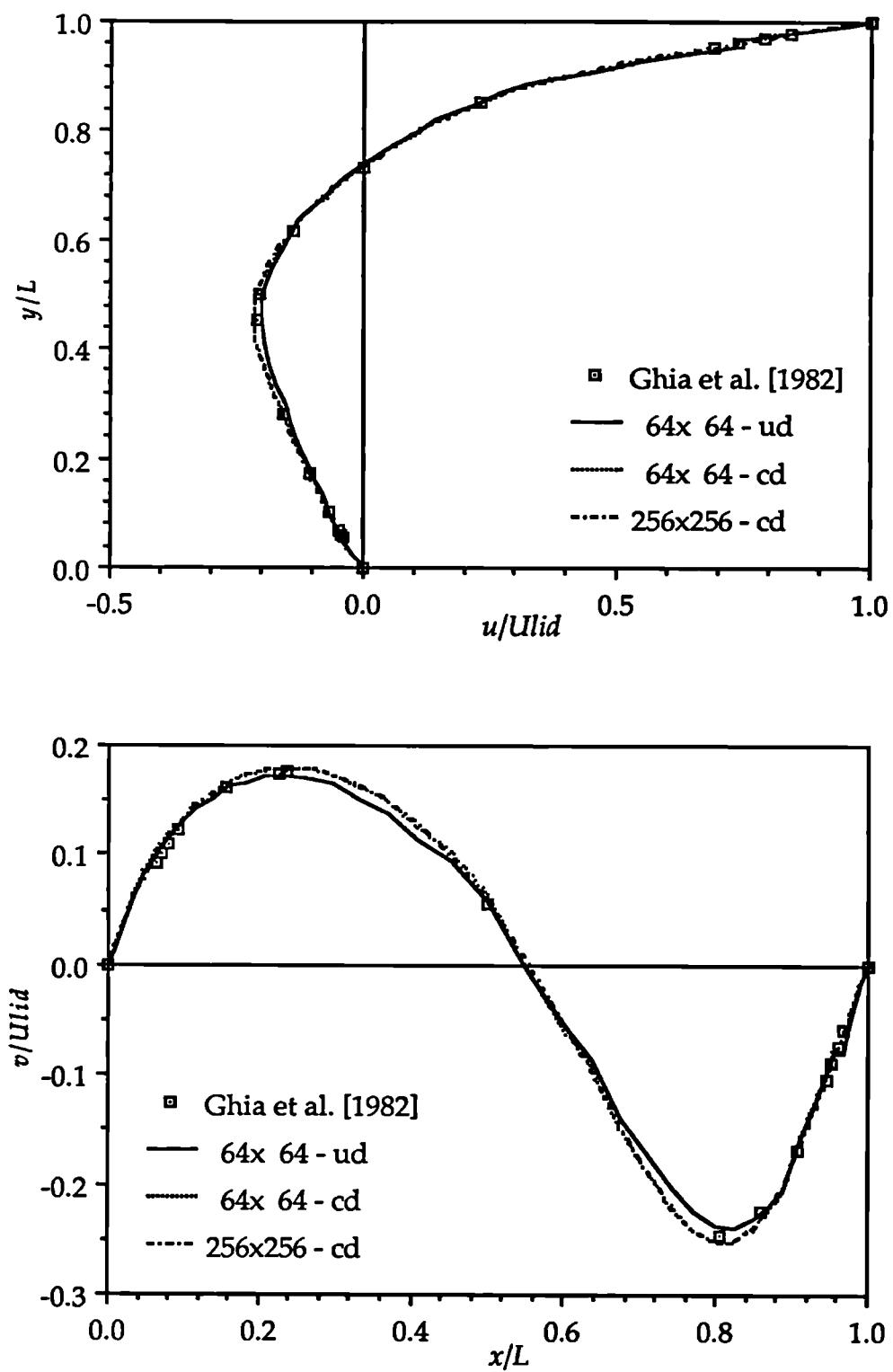


Figure 6.5 Comparison of u and v velocity component profiles along the vertical and horizontal line through the geometric centre at $\text{Re}=100$.

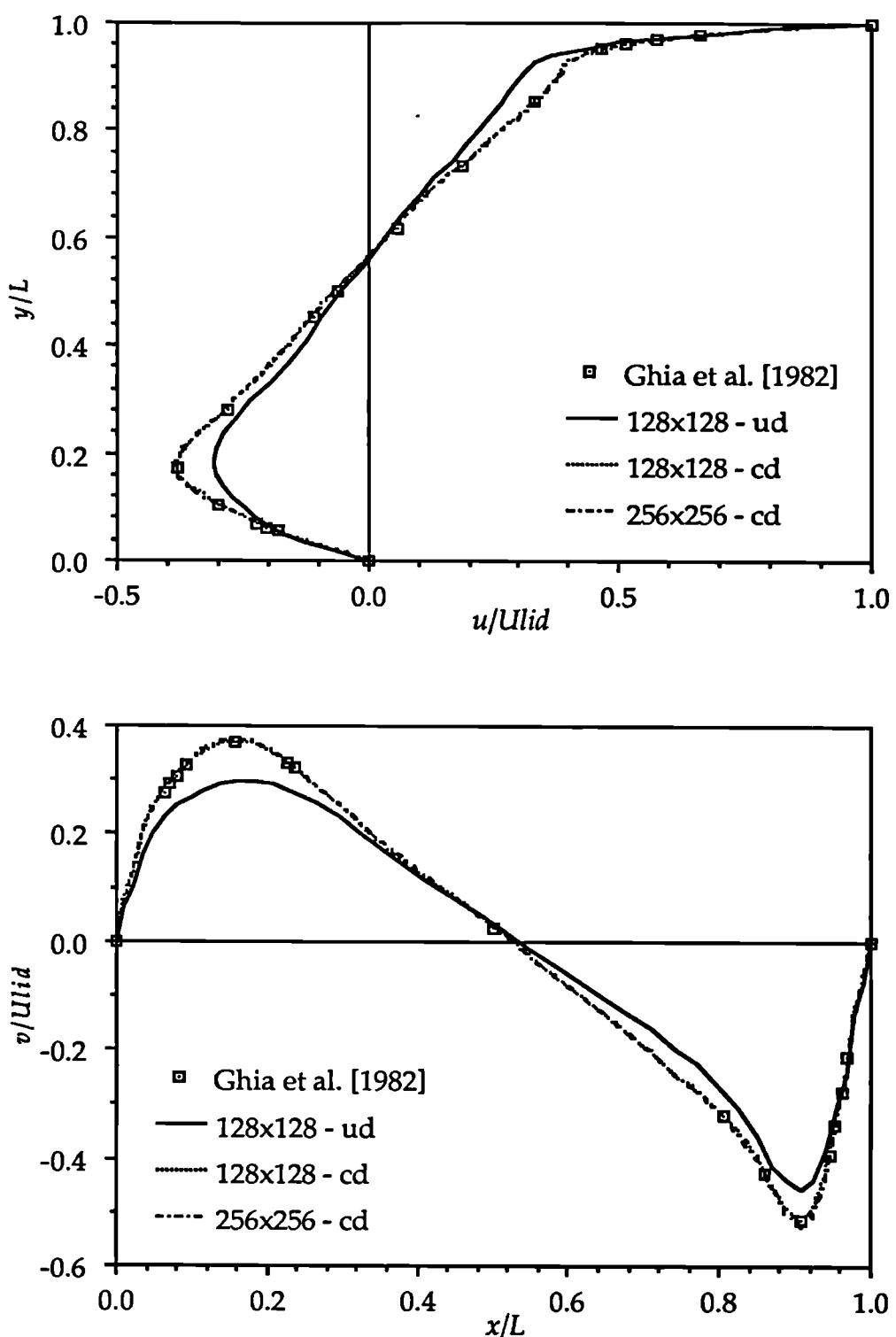


Figure 6.6 Comparison of u and v velocity component profiles along the vertical and horizontal line through the geometric centre at $\text{Re}=1000$.

Central Differencing Scheme					
Grid	8x8	16x16	32x32	64x64	128x128
Number of Fine Grid Iterations					
SGM	43	50	147	552	1984
PGM	40	38	89	212	381
FMG*	43	40	38	35	35
FMG	40	37	35	32	32
Computing Time (CPU)					
SGM	0.65	3.14	43.50	755.70	14111.47
PGM	0.6	2.30	26.03	299.09	2210.44
FMG*	0.89	3.43	14.60	53.63	225.95
FMG	0.88	3.44	13.22	52.62	218.79
SGM/FMG CPU					
	0.73	.91	3.29	14.36	64.50

Upwind Differencing Scheme					
Grid	8x8	16x16	32x32	64x64	128x128
Number of Fine Grid Iterations					
SGM	39	42	127	486	1913
PGM	37	36	80	264	814
FMG*	39	39	38	37	39
FMG	37	37	35	32	32
Computing Time (CPU)					
SGM	0.57	2.58	37.96	693.51	13328.60
PGM	0.56	2.24	25.10	399.58	5391.90
FMG*	0.81	3.42	14.64	52.92	259.66
FMG	0.86	3.55	13.38	52.45	218.63
SGM/FMG CPU					
	0.66	0.73	2.77	13.22	60.96

Table 6.2 Fine grid iteration and computing time required to obtain convergence for Re=100

Central Differencing Scheme						
Grid	8x8	16x16	32x32	64x64	128x128	256x256
Number of Fine Grid Iterations						
SGM	81	119	151	343	998	2960
PGM	76	115	174	298	514	911
FMG*	71	101	106	76	58	46
FMG	64	65	63	40	31	25
Computing Time						
SGM	3.85	24.22	132.76	1477.24	20976.73	392240.2
PGM	3.66	25.57	183.68	1511.16	13823.00	93788.9
FMG*	3.95	26.67	126.40	380.73	1128.00	3868.1
FMG	3.81	19.30	80.06	219.84	658.53	1966.7
SGM/FMG CPU						
	1.01	1.25	1.66	6.72	31.85	199.44

Upwind Differencing Scheme						
Grid	8x8	16x16	32x32	64x64	128x128	256x256
Number of Fine Grid Iterations						
SGM	43	48	93	248	836	2918
PGM	43	46	74	193	625	1998
FMG*	43	45	52	59	66	88
FMG	37	37	37	35	33	38
Computing Time						
SGM	1.91	8.97	81.74	1028.99	18587.11	385668.6
PGM	1.92	8.76	63.85	835.78	15154.83	218361.5
FMG*	2.40	11.52	57.58	286.49	1271.27	7657.8
FMG	2.14	9.45	43.75	181.45	714.15	3323.5
SGM/FMG CPU						
	0.89	0.95	1.87	5.67	26.03	116.04

Table 6.3 Fine grid iteration and computing time required to obtain convergence for $Re=1000$

Tables 6.2 and 6.3 present the total number of fine grid iterations and CPU time for the SGM and PGM calculations and the non-adaptive FMG and FMG* methods. The calculations were performed on a SGI Challenge machine with -O2 optimisation. Determination of the optimum combination of relaxation factors (see Sections 3.6.3 and 3.8) was not performed since it is problem-dependent: thus such an analysis is neither general nor of great practical interest. An analysis of parameters α_p and α_v for the driven cavity test case is given for example by Barcus et al. [1987]. The convergence tolerance for the overall calculation was 10^{-4} . The Gauss-Seidel solver was used for relaxation of the linearised momentum equations and the Incomplete Cholesky preconditioned CG solver (ICCG) for the pressure correction equation.

From Figure 6.7, it can be seen that the convergence rate of the 'standard' SIMPLE algorithm (Sections 3.6.3 and 3.8) without multigrid decreases dramatically at the beginning of iterative process, and after the first few outer (SIMPLE) iterations becomes very slow indeed. For the SGM the number of fine-grid iterations to convergence increases with the number of computational points (Tables 6.2 and 6.3) with an exponent between 0.77 and 0.99 for Re=100 and 0.58 and 0.90 for Re=1000. The CPU time increases with an exponent between 1.89 and 2.13 for Re=100 and 1.73 and 2.18 for Re=1000.

The FMG asymptotic convergence rate (effectively being the negative of the slope of a curve on Figure 6.7) does not change with the number of computational points, so that the number of fine grid iterations remains almost the same (actually decreases for Re=1000 with an average exponent of -0.16) while the computing time increases linearly with the number of computational points. These results are in complete accordance with those of Vanka [1986], Gaskell et al. [1987] and Barcus et al. [1987]. The asymptotic convergence factors (defined by equation (4.23)) corresponding to the SIMPLE algorithm and the given problem were worked out from the slopes of the convergence curves and presented on Figure 6.8. The asymptotic value of the convergence factor for FMG remains almost constant, with an average value of 0.832 for Re=100. This means that the ratio of residuals between two consecutive iteration is 0.832 or that the average residual will be reduced by an order of magnitude in about 12.52

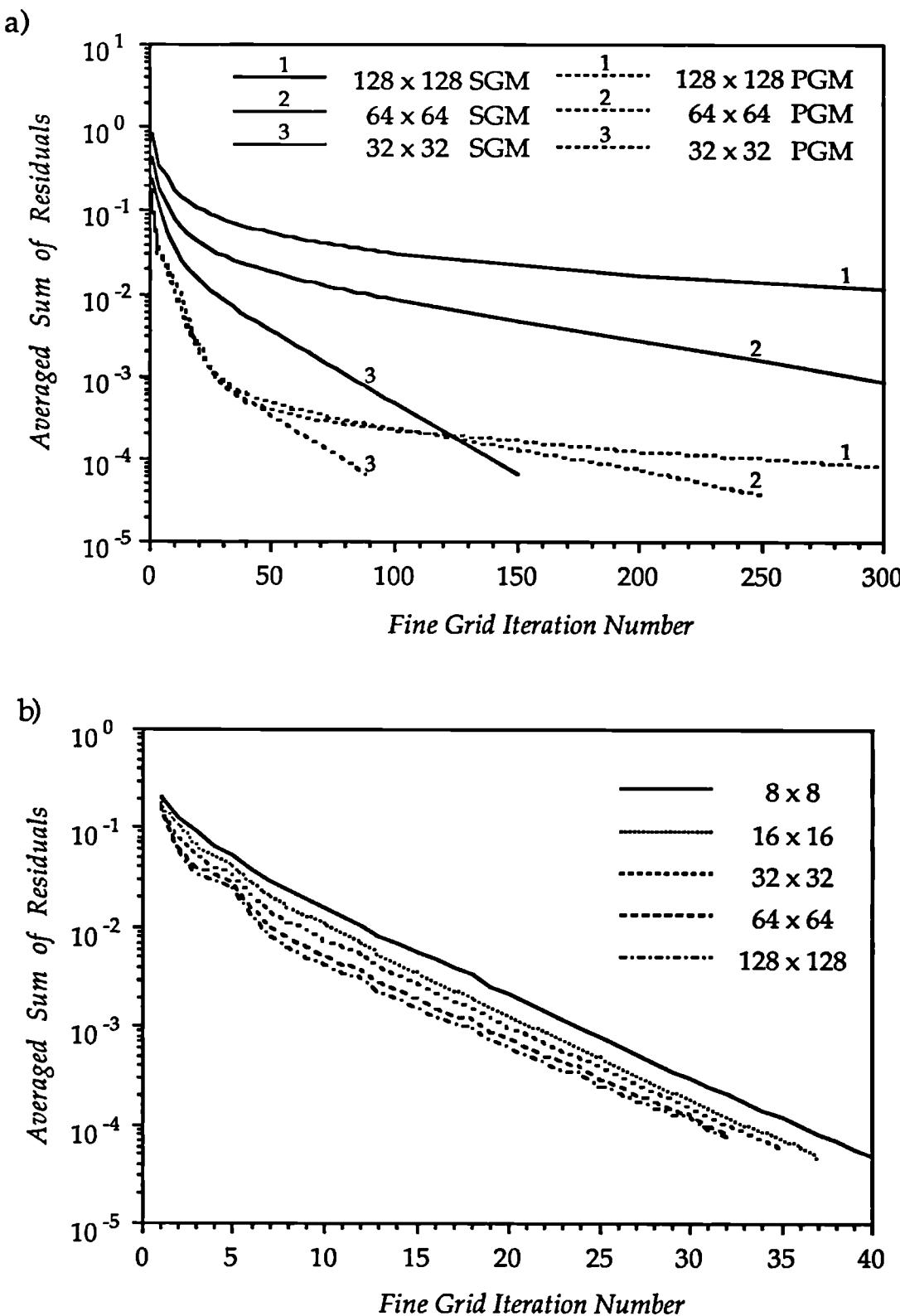


Figure 6.7 Single grid (a) and multigrid (b) convergence histories for $Re=100$

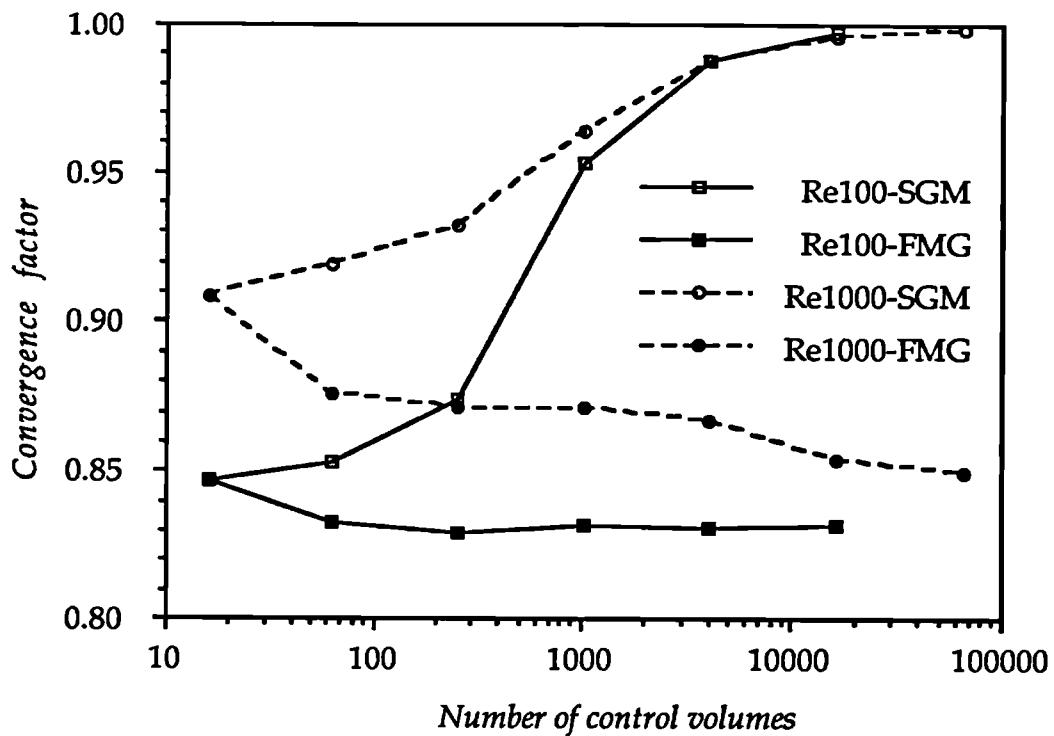


Figure 6.8 Asymptotic value of convergence factor

iterations. The maximum asymptotic value of the convergence factor for SGM is 0.997 for $Re=100$, which means that around 766.38 iterations are necessary to reduce the average residual by an order of magnitude. The conclusions are similar for the $Re=1000$ calculations.

Comparing the results of the SGM and PGM, one can appreciate the importance of a good initial guess, which is provided by the nested iteration and is a part of full multigrid algorithm. The effects of extrapolation of the coarse grid solution to the next finer mesh can be seen from Figure 6.7.a. The 'asymptotic' slopes of the curves presented on Figure 6.7.a, for both SGM and PGM, are the same. The only difference is that the amplitude of the smooth error components, at the initial stage, is much smaller for the PGM, since it starts with a better initial guess. For $Re=100$ the ratio of CPU times for the SGM and PGM is 6.38. The work necessary to get a good initial guess, in this case, is not a very large overhead. If the CPU times on all grids necessary to get an initial guess on the finest grid, are summed up, the ratio of CPU times for the SGM and PGM is 5.56. The PGM is less impressive for the cavity flow at $Re=1000$,

where the saving in CPU time relative to the SGM method is 4.18. The reason for this is that the solution is less smooth for $Re=1000$, and the interpolated coarse-grid solutions are too inaccurate, lacking important fine-scale details to be effective starting fields on a fine grid. This has been observed by Lien [1992] as well, and is much more significant for turbulent flow calculations.

Very similar conclusions can be drawn comparing the figures for FMG and FMG* given in Tables 6.2 and 6.3. For the smooth variation of dependent variables at $Re=100$, the different interpolation practices used for prolongation do not make a big difference in the convergence rate of the multigrid iteration. However, for $Re=1000$ the first-order prolongation operator, defined by expression (4.43) and being the part of FMG, leads to the two times faster convergence than the zero-order one, which is implemented in FMG* and is given by equation (4.42).

6.2.2 Multigrid with Adaptive Local Grid Refinement

In this section results obtained with local grid refinement are presented. To identify regions requiring refinement, use was made of both criteria described in Section 5.4, and results were compared with each other. The criteria based on Richardson extrapolation (Section 5.4.1) is called C1, and the novel one based on the approximation of the truncation error using the polynomial fit (Sections 5.4.2 and 5.4.3), is denoted as C2.

Figure 6.9 presents the 'exact' error Eu and Ev for the u and v velocity components for Reynolds number 100, obtained by comparing the solution on the finest grid 256x256 (being considered as the exact one) with the solutions on grids 8x8 to 64x64.

The solution and tau error estimations based on C1 and C2 are plotted in Figures 6.10 to 6.12. The data presented was normalised in the manner described in Section 5.5, where the reference value was chosen to be the velocity of the driving lid. Qualitatively, all error estimations behave similarly and in accordance with the solution error presented in Figure 6.9

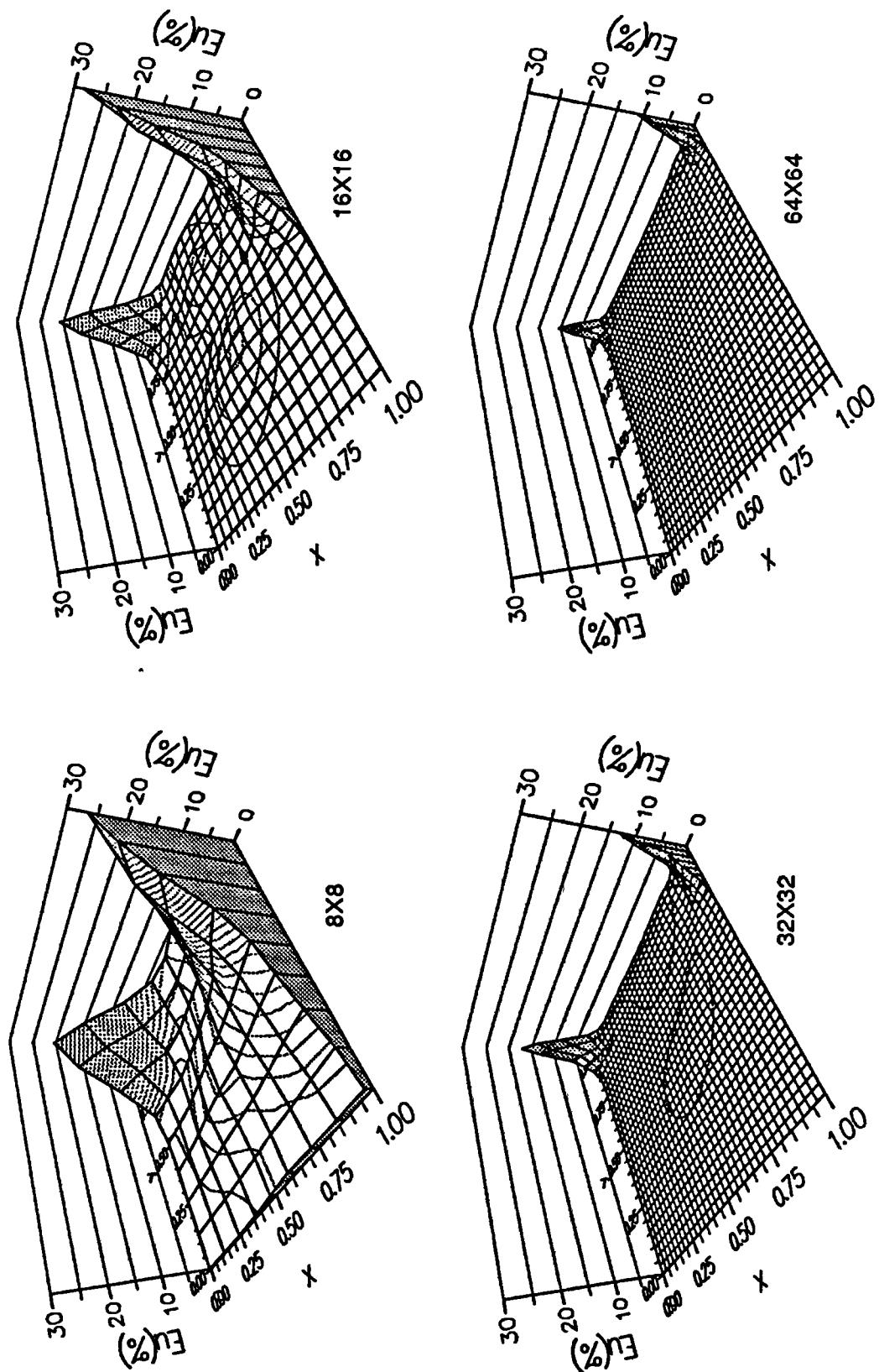


Figure 6.9 'Exact' error on different grids for u (E_u) and v (E_v) velocity vector component at $Re=100$

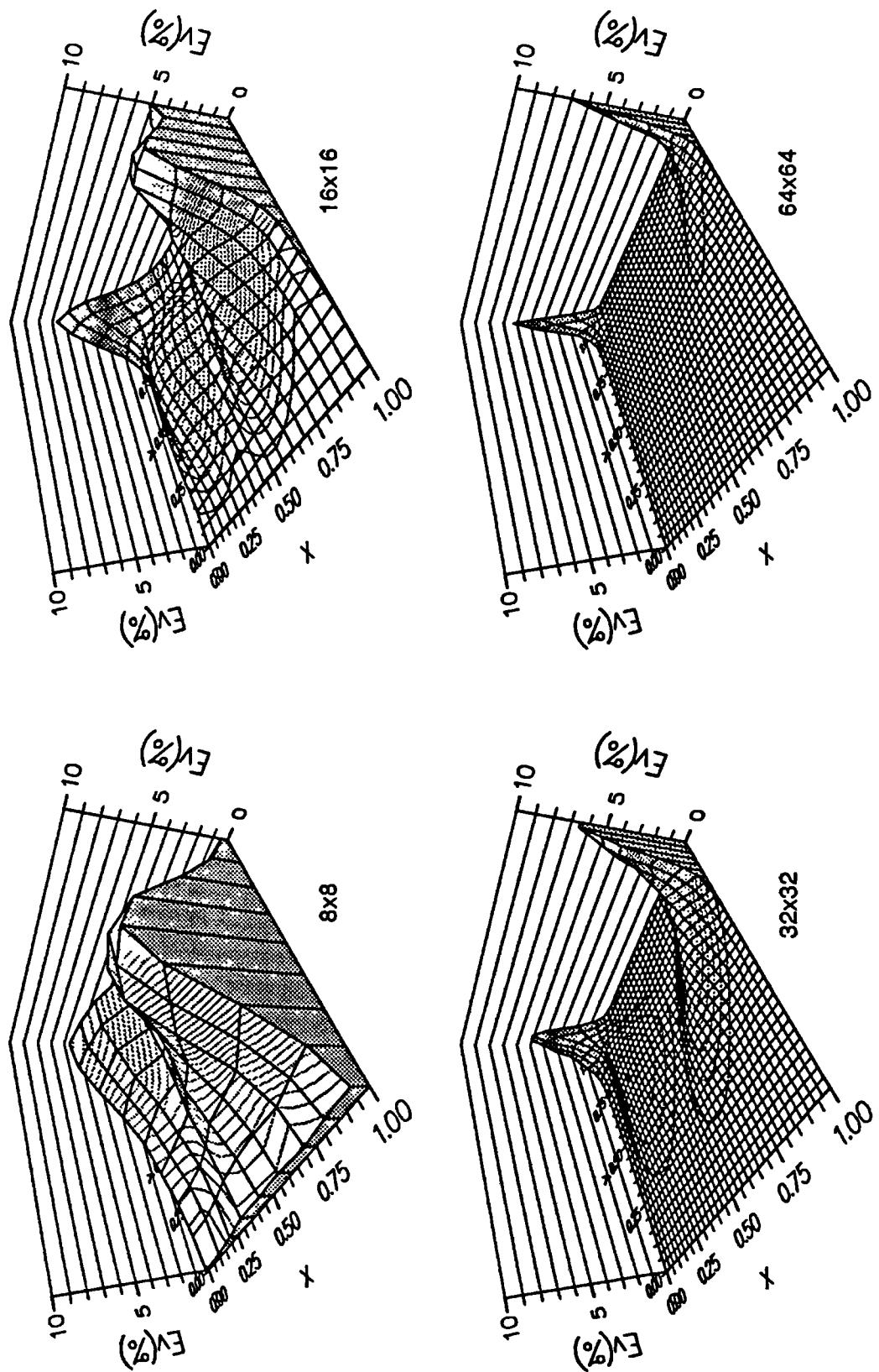


Figure 6.9 (continued)

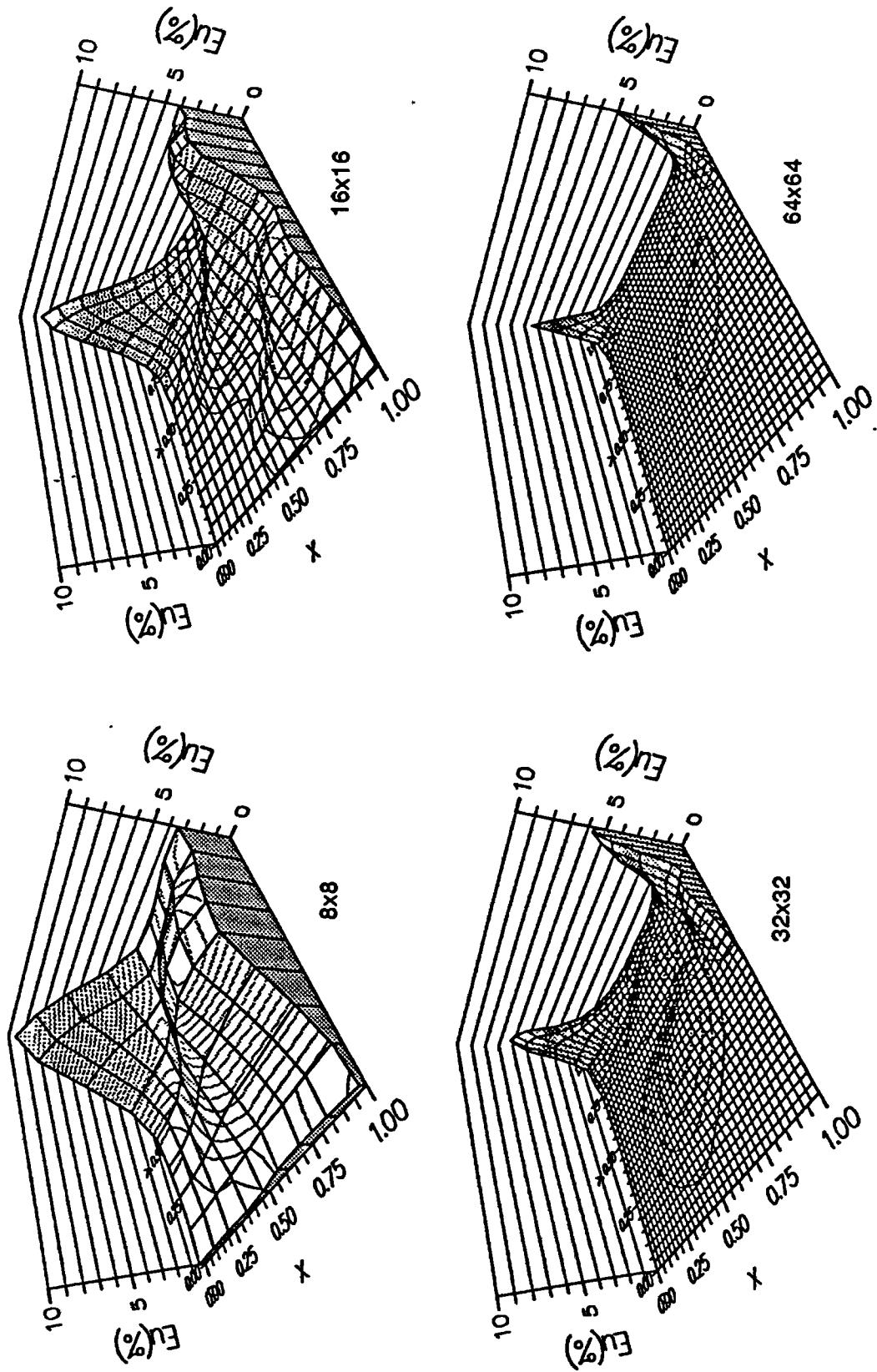


Figure 6.10 Error estimation C1 on different grids for u (E_u) and v (E_v) velocity vector components at $Re=100$

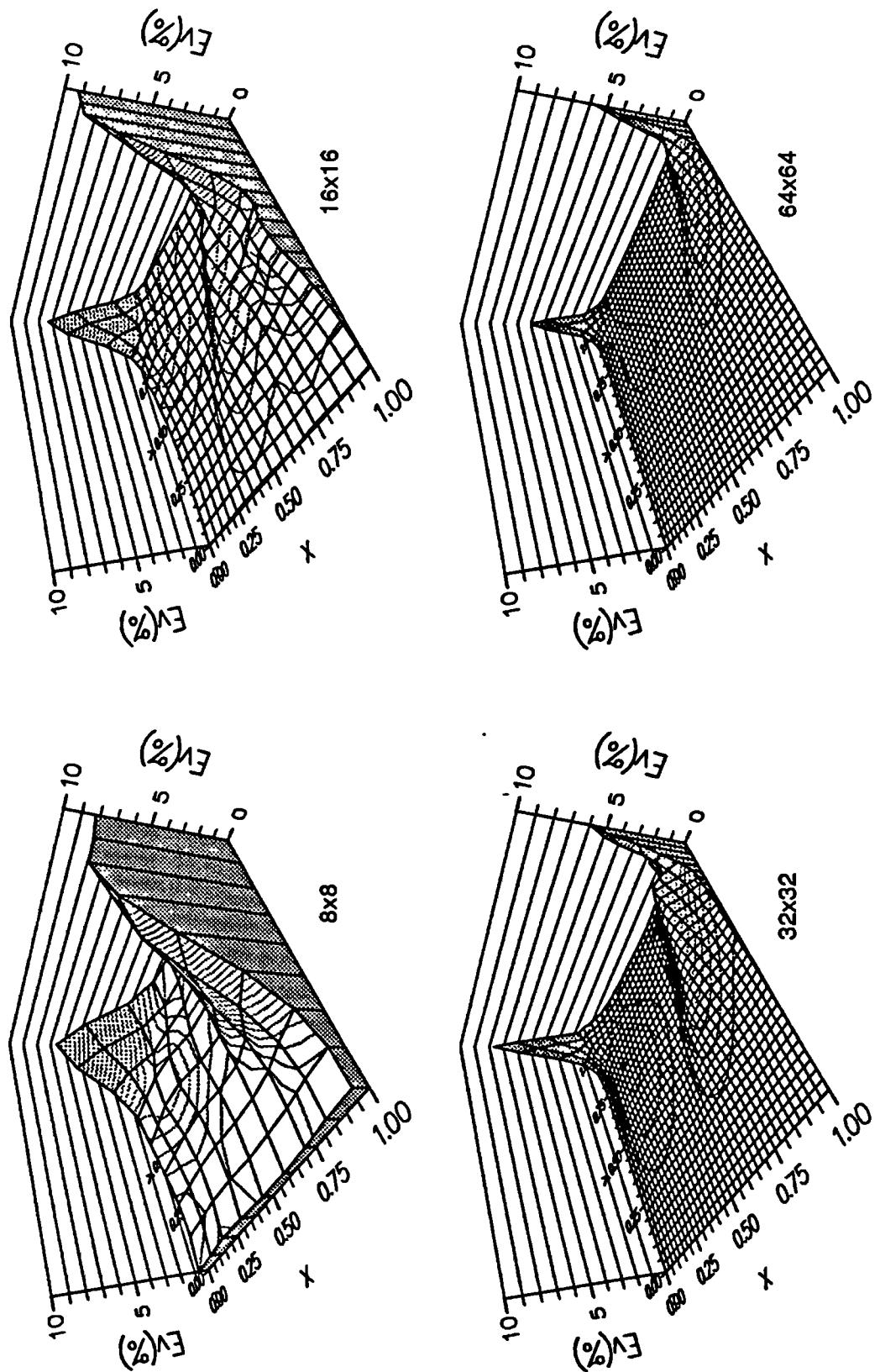


Figure 6.10 (continued)

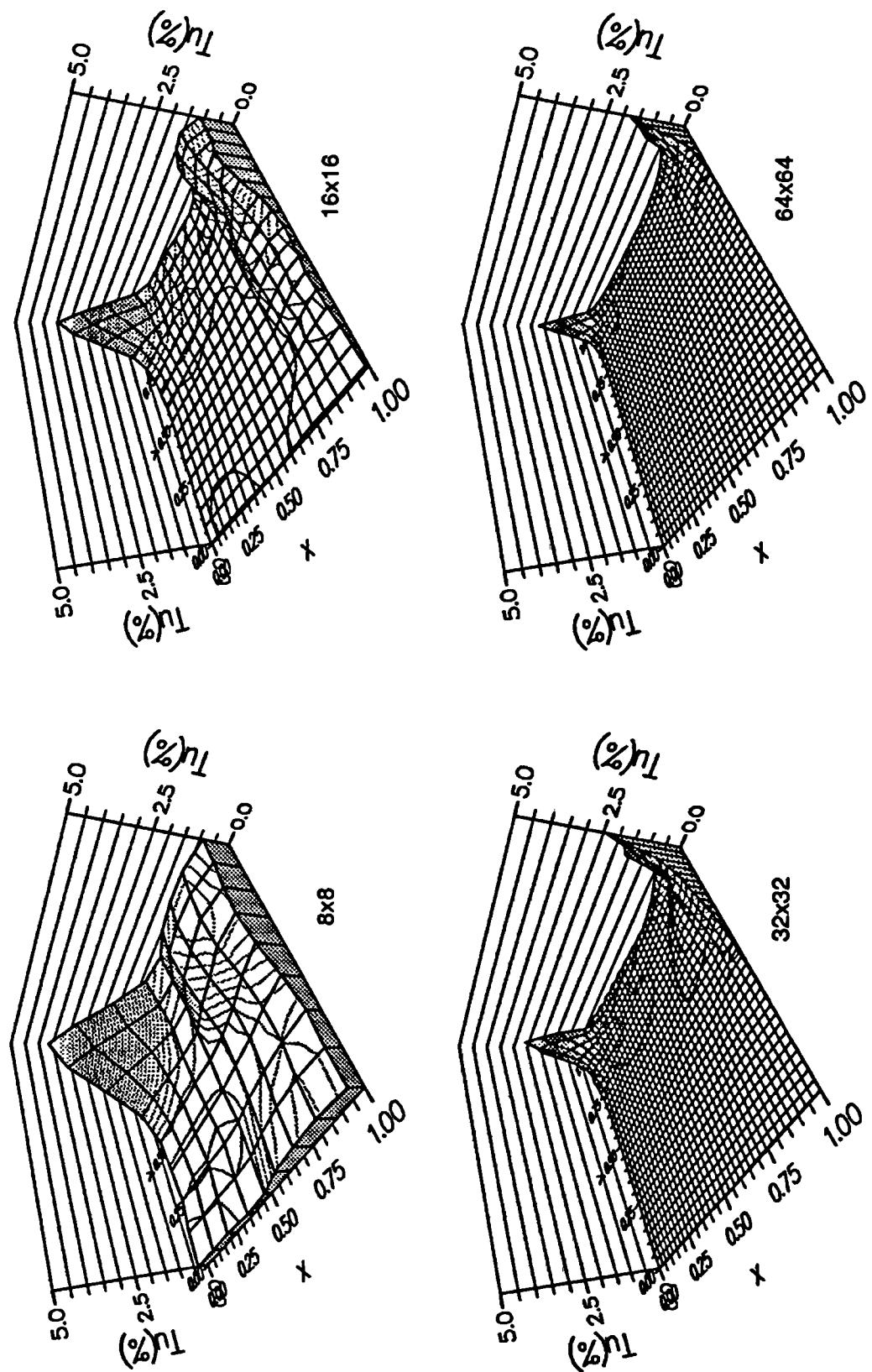


Figure 6.11 Tau error C_1 on different grids for u (T_u) and v (T_v) velocity vector components at $Re=100$

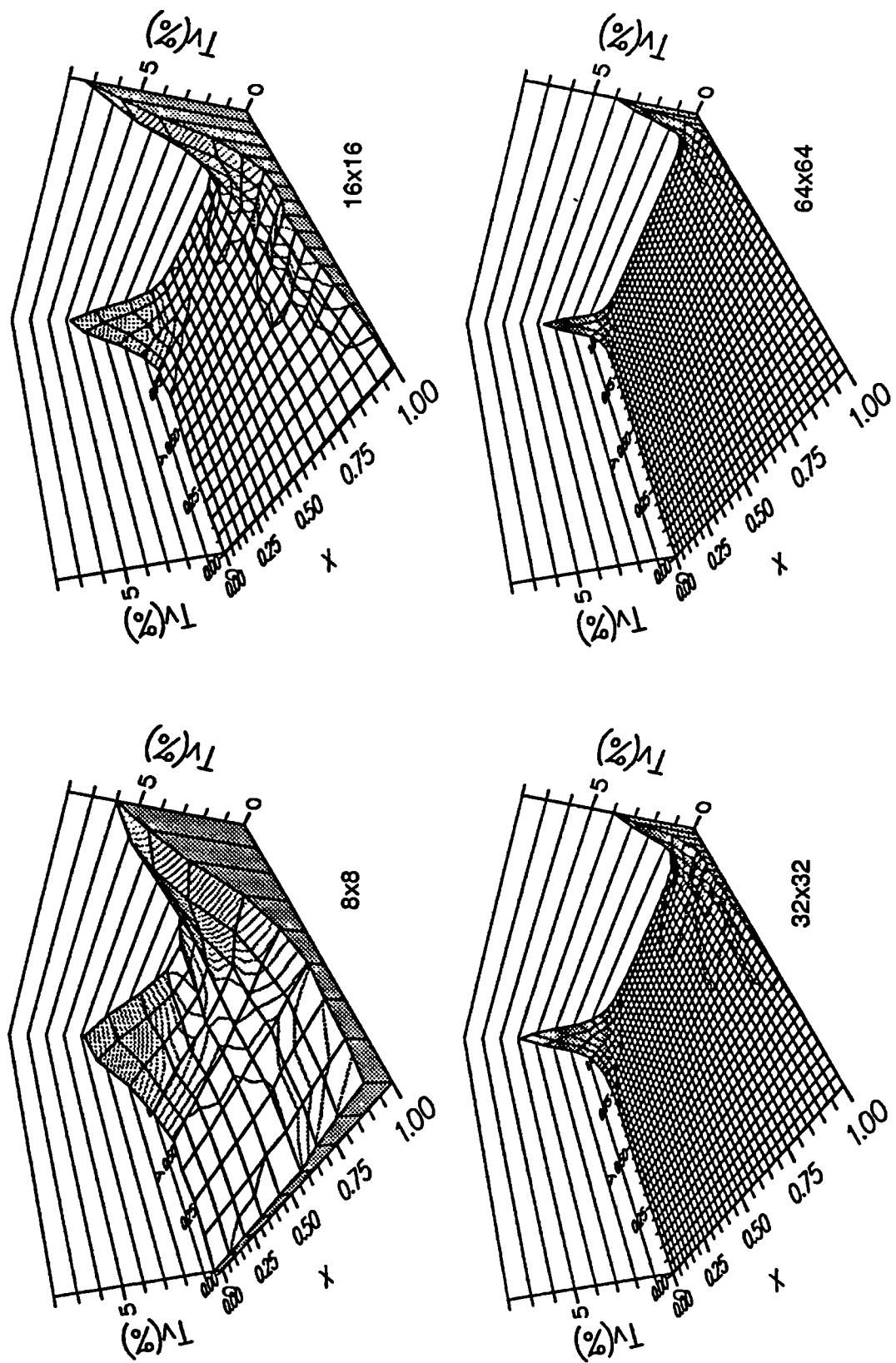


Figure 6.11 (continued)

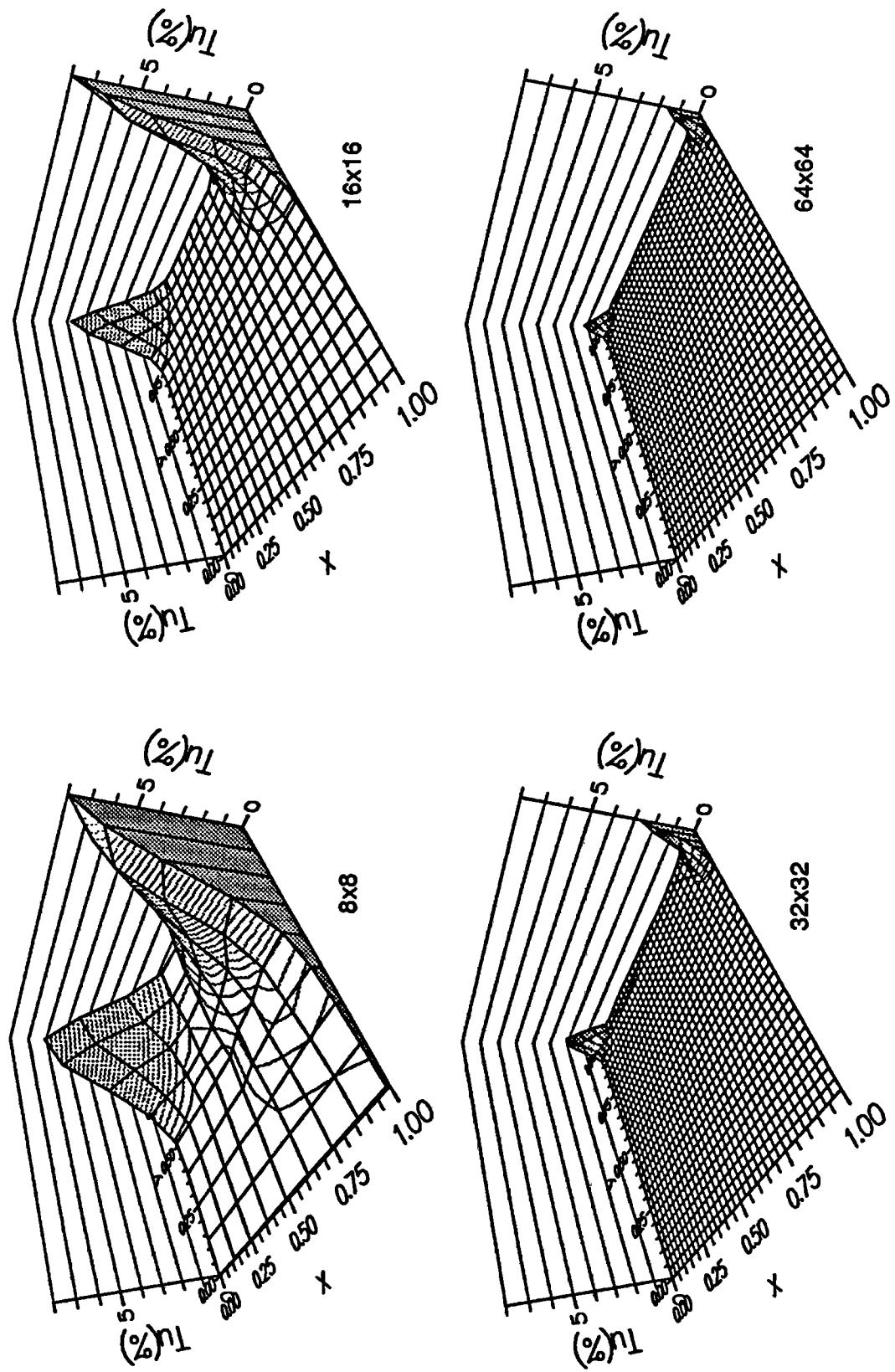


Figure 6.12 Tau error C_2 on different grids for u (T_u) and v (T_v) velocity vector components at $Re=100$

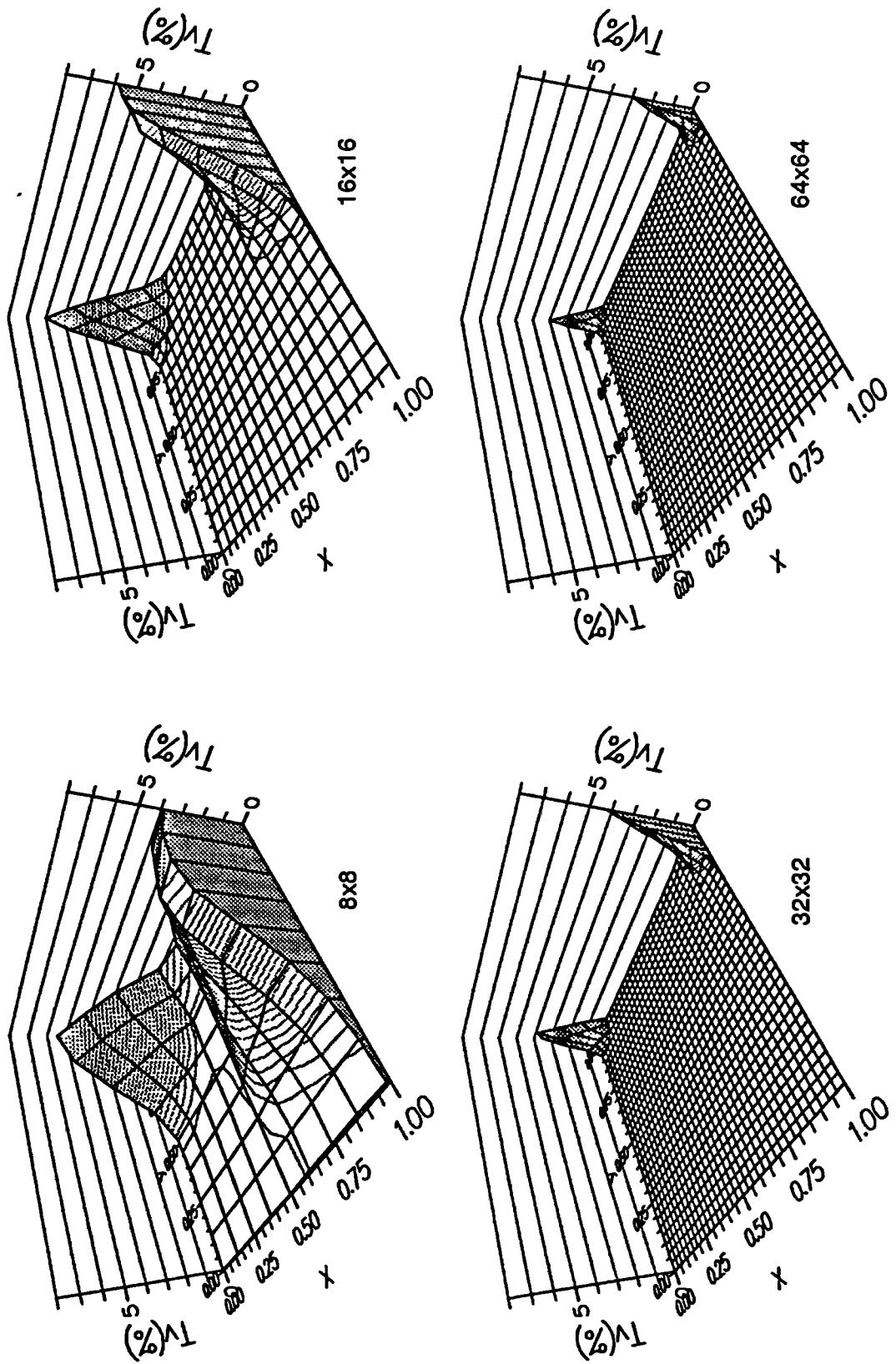


Figure 6.12 (continued)

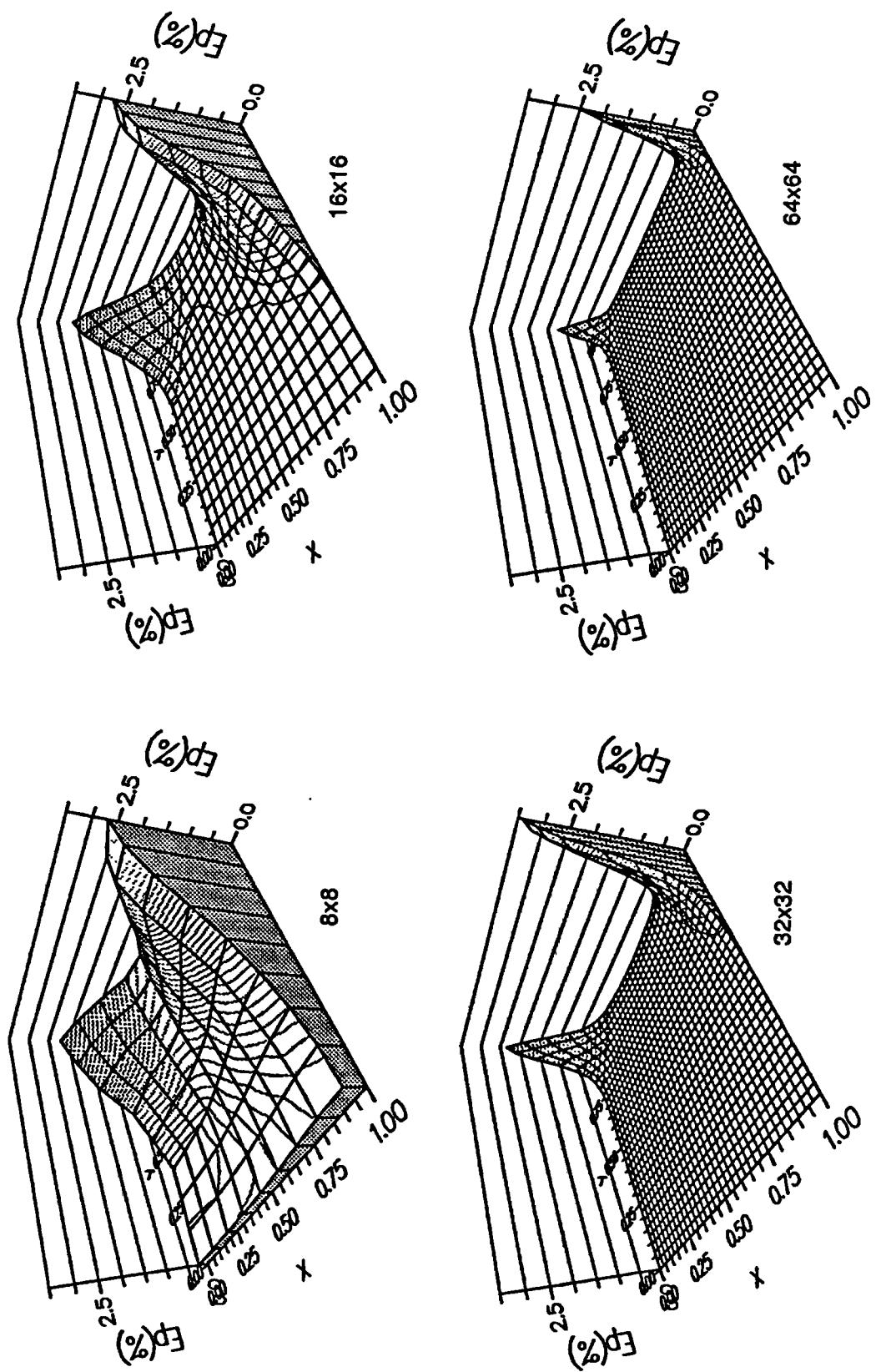


Figure 6.13 Error C2 for pressure (Ep) on different grids at $Re=100$

and general expectations. The larger solution and truncation errors are confined to the regions adjacent to the two top corners ($x=0, y=1$) and ($x=1, y=1$), where driving lid meets the fixed walls, the flow sharply changes direction, and large gradients of all dependent variables exist.

The estimate of the error introduced by approximating the space variation of pressure is given in Figure 6.13. The distribution of this error is consistent with the estimates for other variables, indicating clearly that the largest variation of pressure, and so the largest sources of the error are present in the two top corners. These results are qualitatively in good agreement with those given by Smith [1991].

The quantitative behaviour of the solution error estimations can be obtained from Figure 6.14, where average and maximum solution errors and their effectivity indices (the ratio of an error estimate and the exact error as defined by expression (5.3)) for velocity magnitude are plotted against the number of control volumes used in the calculation. Four sets of data are presented: the exact error, the error estimate obtained using Richardson extrapolation (C1 criterion), the error estimate based on the estimation of the truncation error (C2 criterion), and the error estimate obtained solving the transport equation for the solution error (expression 5.8) using the tau error estimate C2. The weighting factors for averaging were obtained dividing cell volumes by the volume of the domain.

Both error estimations, C1 and C2, underpredict the average and maximum error. However, the absolute differences diminish and the effectivity index approaches one as the mesh is refined. The average error estimation based on Richardson extrapolation has a better effectivity index than the other two techniques. Solution of the error transport equation improves the solution error estimate, as can be seen from Figure 6.14b, where the effectivity index approaches 1 much faster as the number of control volumes increases than the just normalised tau error estimate C2.

The rate of reduction of the maximum error is not as fast as that of the average error. The reason for this might be that the maximum error is isolated in the two top corners where large spatial variations of all

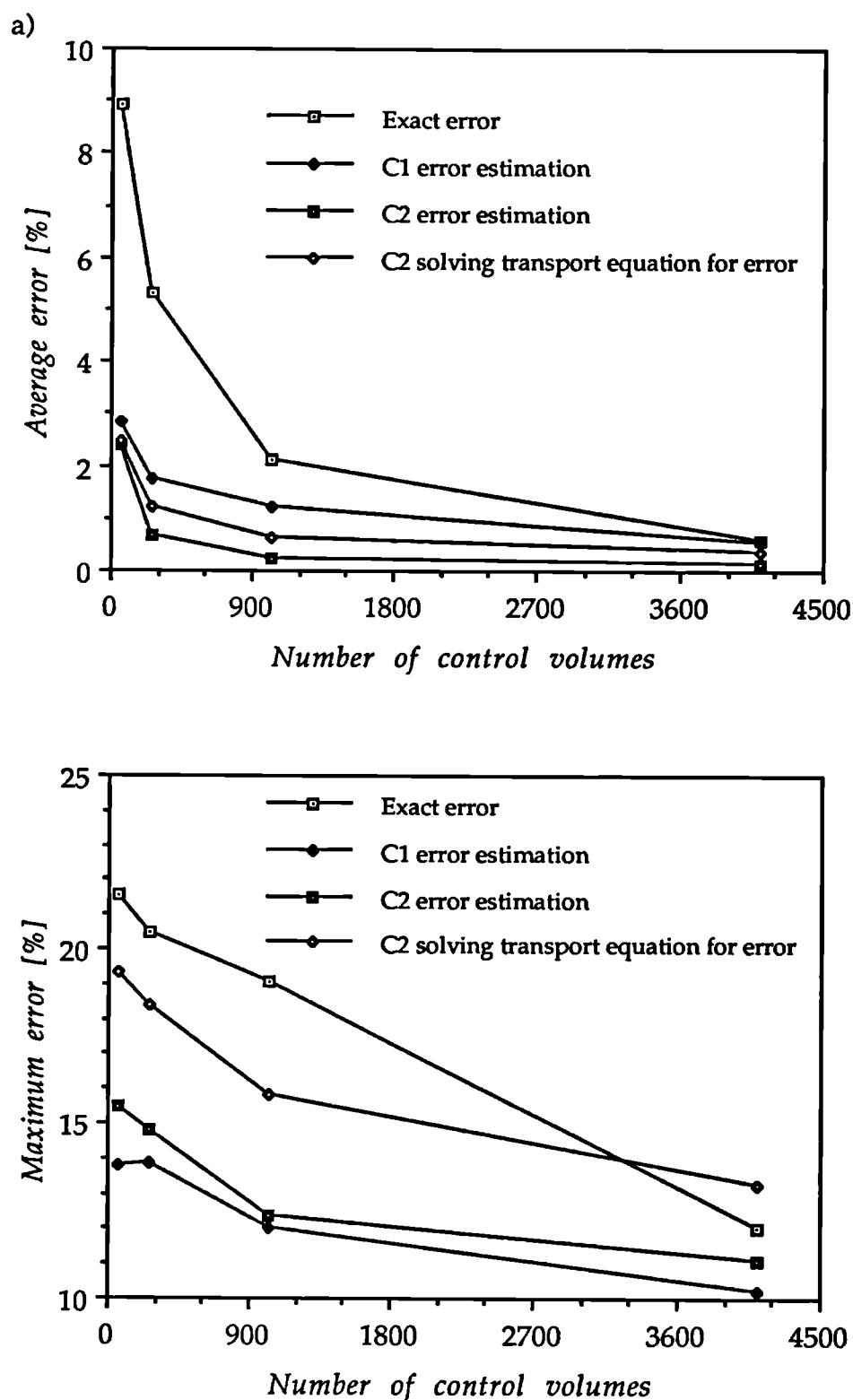


Figure 6.14 Average and maximum error a) and their corresponding effective indexes b) for the driven cavity flow at $Re=100$

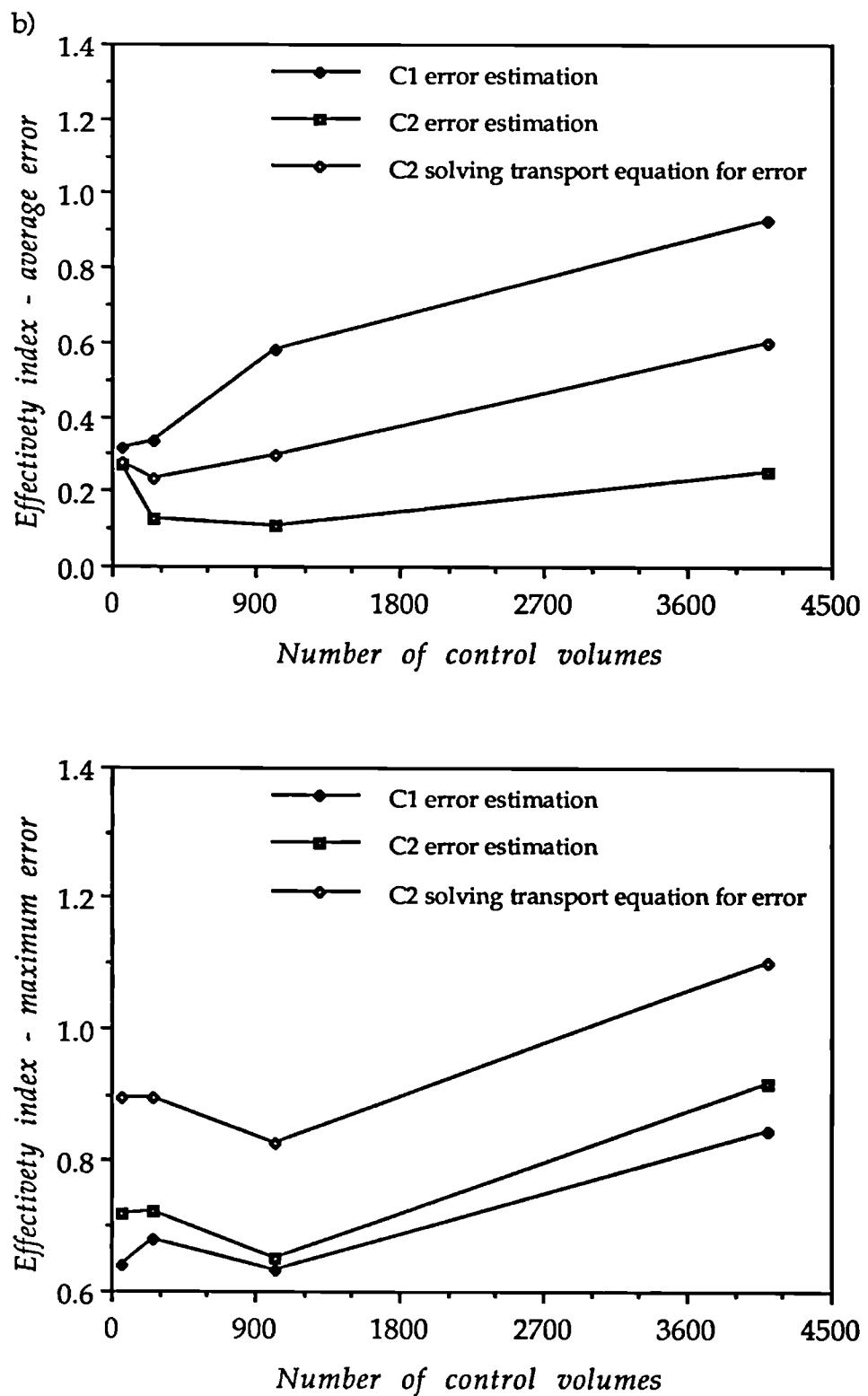


Figure 6.14 (continued)

variables exist and the leading terms of the truncation error are not the dominant components. The reduction in the solution error with increasing mesh resolution is consequently not as large as in other regions, where the solution is smooth. However, from Figure 6.14 it seems that the estimated leading terms of the truncation error approximate well the maximum solution error.

The main effects of solving the transport equation for the solution error C_2 is the increase of the average and maximum solution error estimates, improving the solution error estimate. These increases are to be expected, since we recall, the normalisation of the tau error estimate is based on the error transport equation (5.8). This normalisation is performed in such a way that the influence of the solution errors at the neighbouring points are neglected, and the tau error is divided by only the central coefficient (expressions (5.26) & (5.28)). By solving the error transport equation, the solution error at a certain point is influenced by the distribution of the solution error in its neighbourhood. If the estimated tau error at that location is zero, the solution error at that cell will be the average of all solution errors surrounding the cell considered. The weighting factors for this averaging are the coefficients that point to these nearest-neighbour cells divided by the central coefficient. If the estimated tau error is not zero, it is divided by the central coefficient and added to the solution error.

If one is interested in a parameter that drives adaptive local grid refinement, solution of the error transport equation based on the estimated tau error does not bring anything qualitatively new, since the estimation of the tau error alone is sufficient. The maximum solution error that is obtained by simple normalisation of the tau error estimate should give a relatively good approximation to the upper limit of the solution error, and this upper limit, rather than the whole solution error field, is a parameter on which the decision about further refinement is based.

All solution error estimates have similar qualitative behaviour (similar slopes for graphs presented in Figure 6.14). Since the average error represents better the global error estimate, one may say that error

estimation based on Richardson extrapolation gives better results. However, this solution error estimate needs solution on two grids and storage for the coarse-grid one. For a multigrid program this is not an important issue, since both are available as a part of general multigrid strategy. For those who want to use error estimation techniques for local-grid refinement without storing the solution and geometrical information of the coarser grid, those additional overheads might be important.

Based on the tau error estimations C1 and C2, the locally refined meshes shown in Figure 6.15 were created adaptively. The regions where the error estimates were larger than the prescribed value of 1%, were extended in all directions by a safety margin d (Section 5.6), which was chosen to be four times the size of the currently-finest cell dimension. Criterion C2 involved the estimation of the error in pressure as well, as described in Section 5.5. The largest control volumes have the spacing of the 8x8 full grid. The finest grid spacing is equivalent to the 64x64 full grid, which was found to produce grid independent velocity profiles along horizontal and vertical planes through the geometric centre of the cavity. The results obtained with these meshes are presented in Figure 6.16. Although the locally refined mesh is composed of 712 CVs, compared to 4096 CVs uniformly-refined mesh, the maximum v velocity component is predicted with solution error of 1.2% of the velocity of the lid. The maximum v velocity on the 16x16 based uniform grid differs by 22% from the solution on the 64x64 grid, showing the considerable improvement due to local refinement.

The normalised solution and tau errors for $Re=1000$ are plotted in Figures 6.17-6.21. In common with the lower Reynolds number case, the largest tau error is confined to the regions surrounding the two top corners. Based on the distribution of the tau errors C1 and C2, the locally-refined meshes shown in Figure 6.22 were created adaptively. They consist of the 16x16 base grid with three levels of local refinement, the finest with a grid spacing equivalent to a full 128x128 grid. The reduction in the number of control volumes due to local refinement is 90% compared to the 128x128 uniform grid, which produced grid-independent velocity profiles. The velocity profiles obtained with these grids are in very good agreement with the grid-independent solution (Figure 6.23), and are much

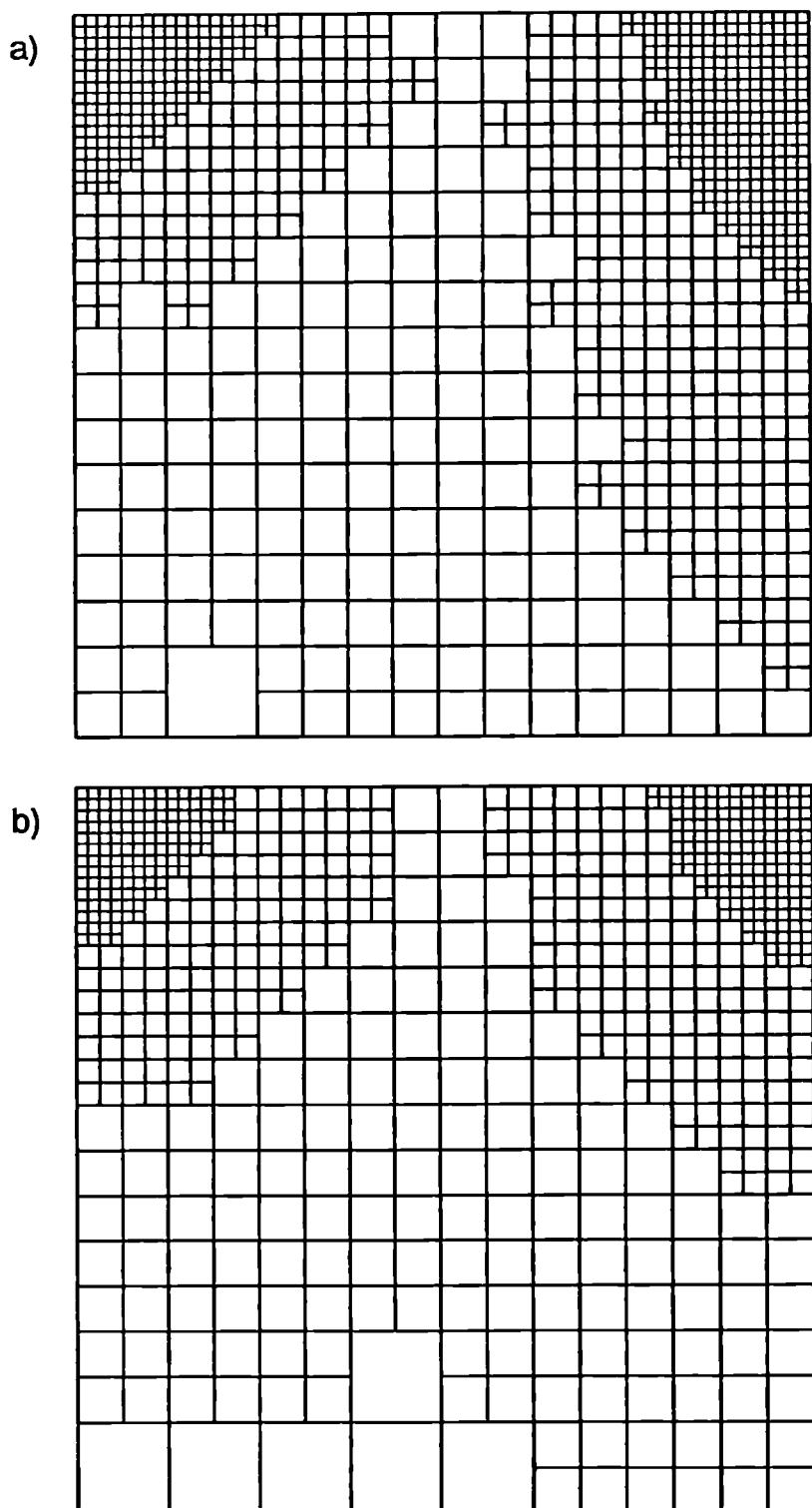


Figure 6.15 Locally refined meshes at $Re=100$ based on criterion C1 (916 CVs) a) and C2 (712 CVs) b).

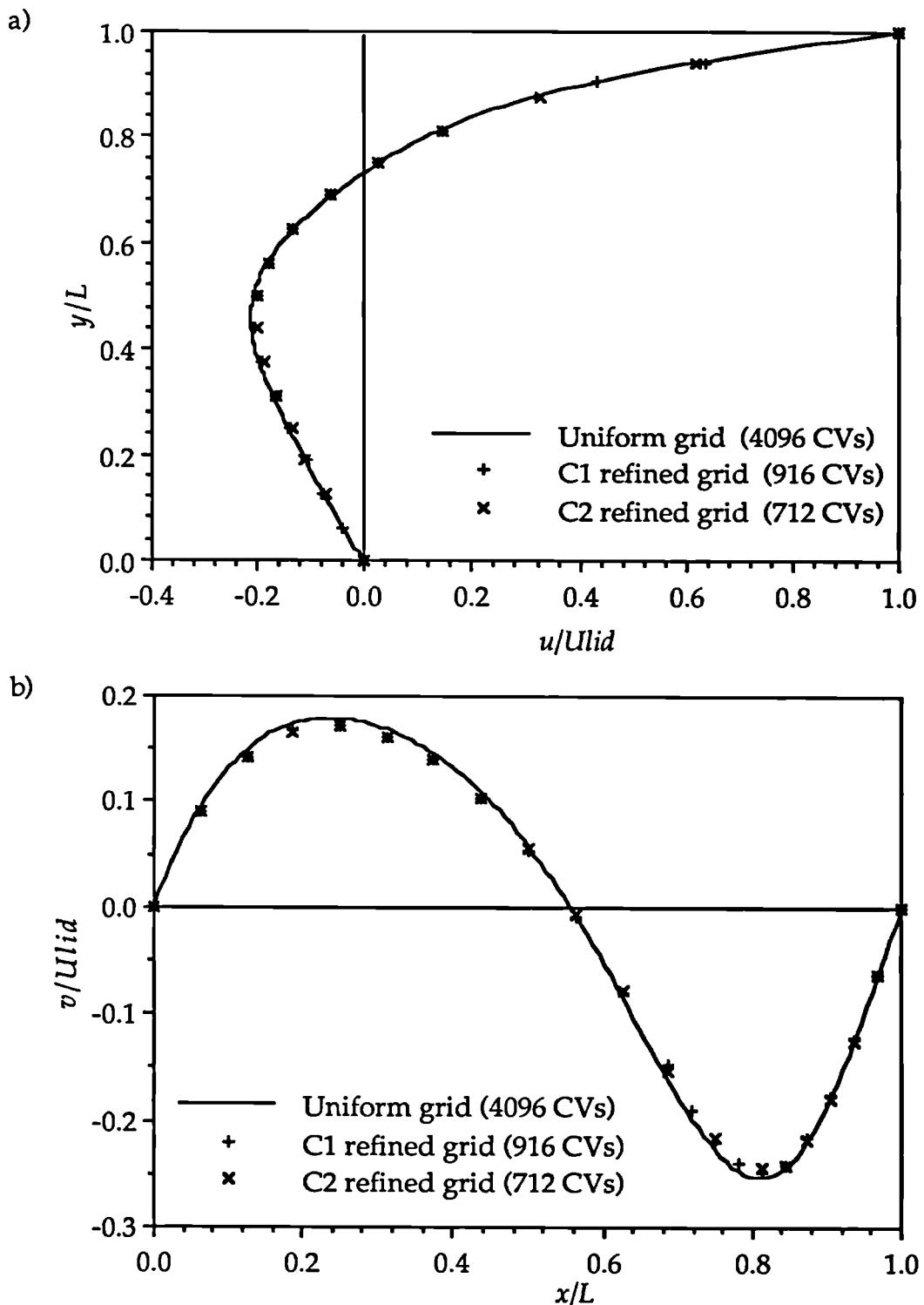


Figure 6.16 Velocity profiles along vertical a) and horizontal b) line obtained on uniform and locally refined grids for $Re=100$.

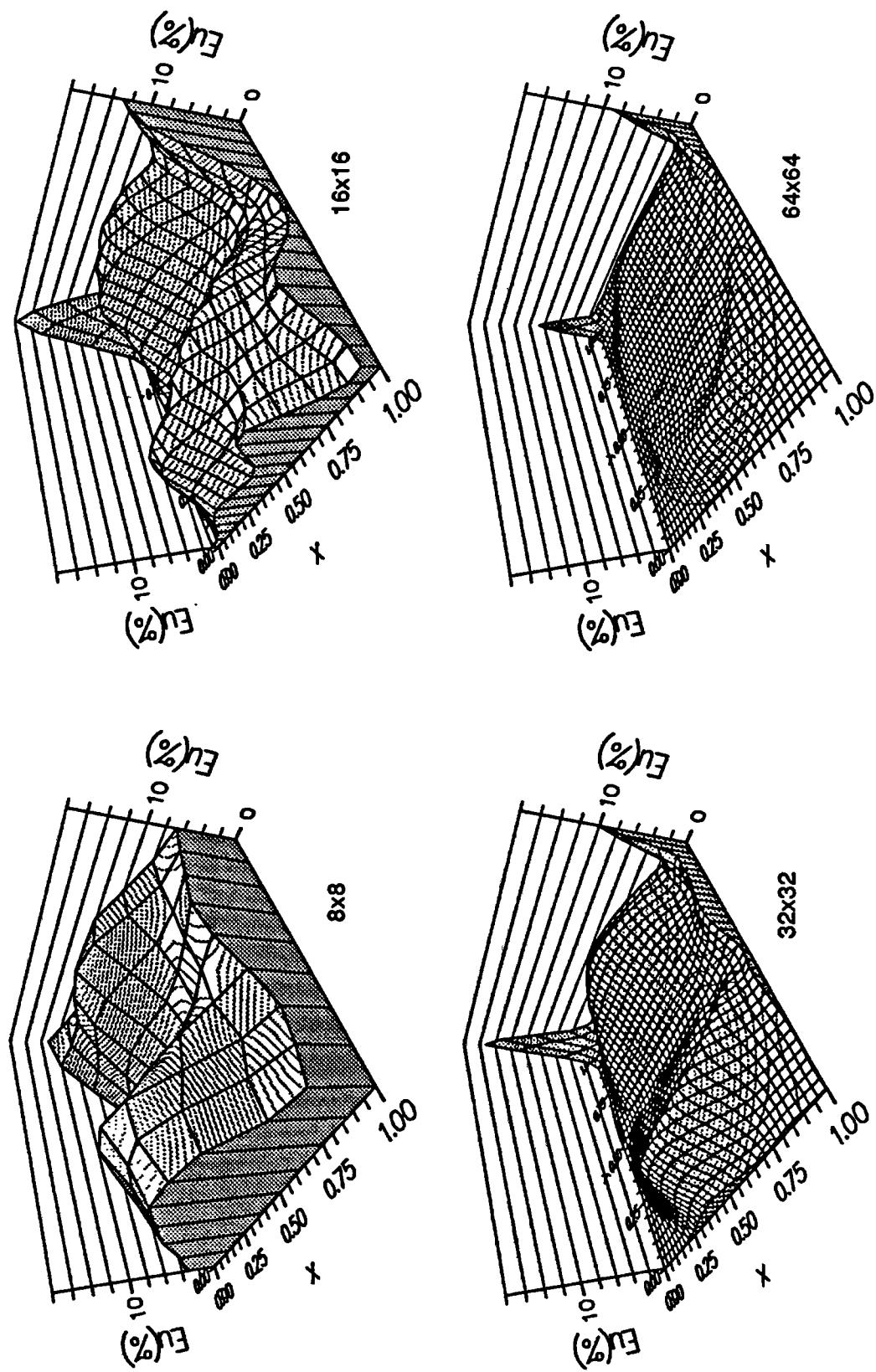


Figure 6.17 'Exact' error on different grids for u (E_u) and v (E_v) velocity vector component at $Re=1000$

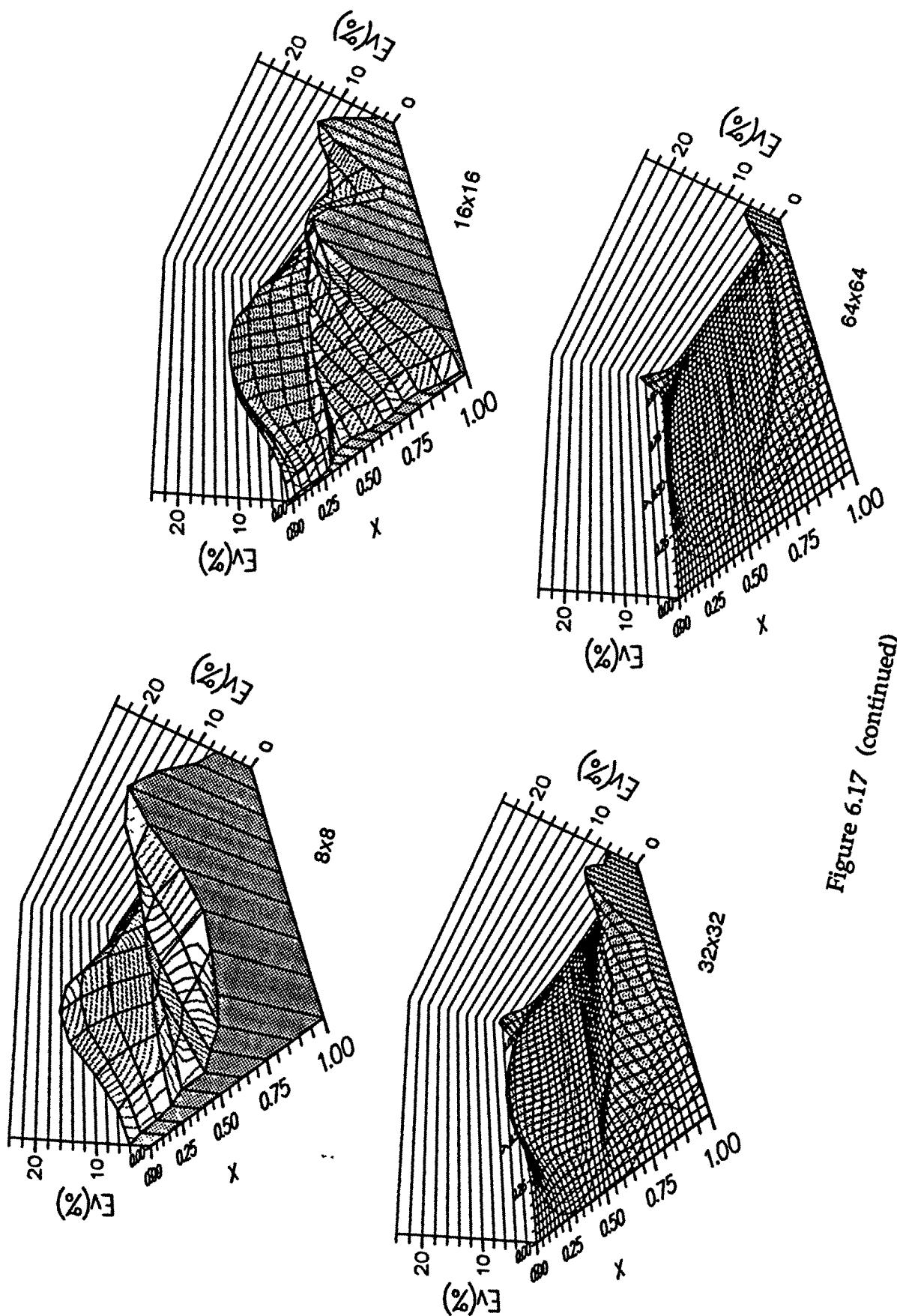


Figure 6.17 (continued)

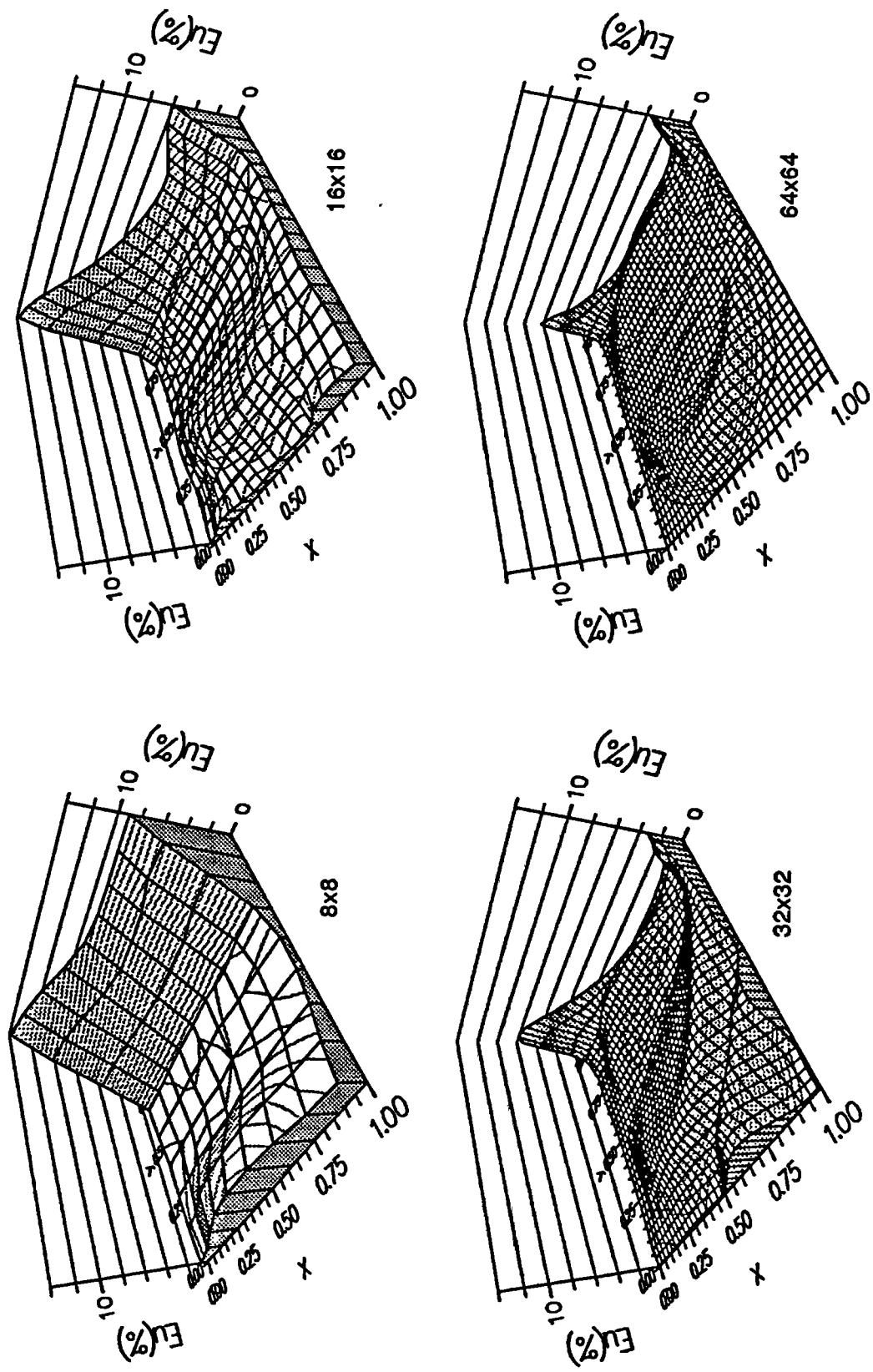


Figure 6.18 Error estimation C1 on different grids for u (E_u) and v (E_v) velocity vector components at $Re=1000$

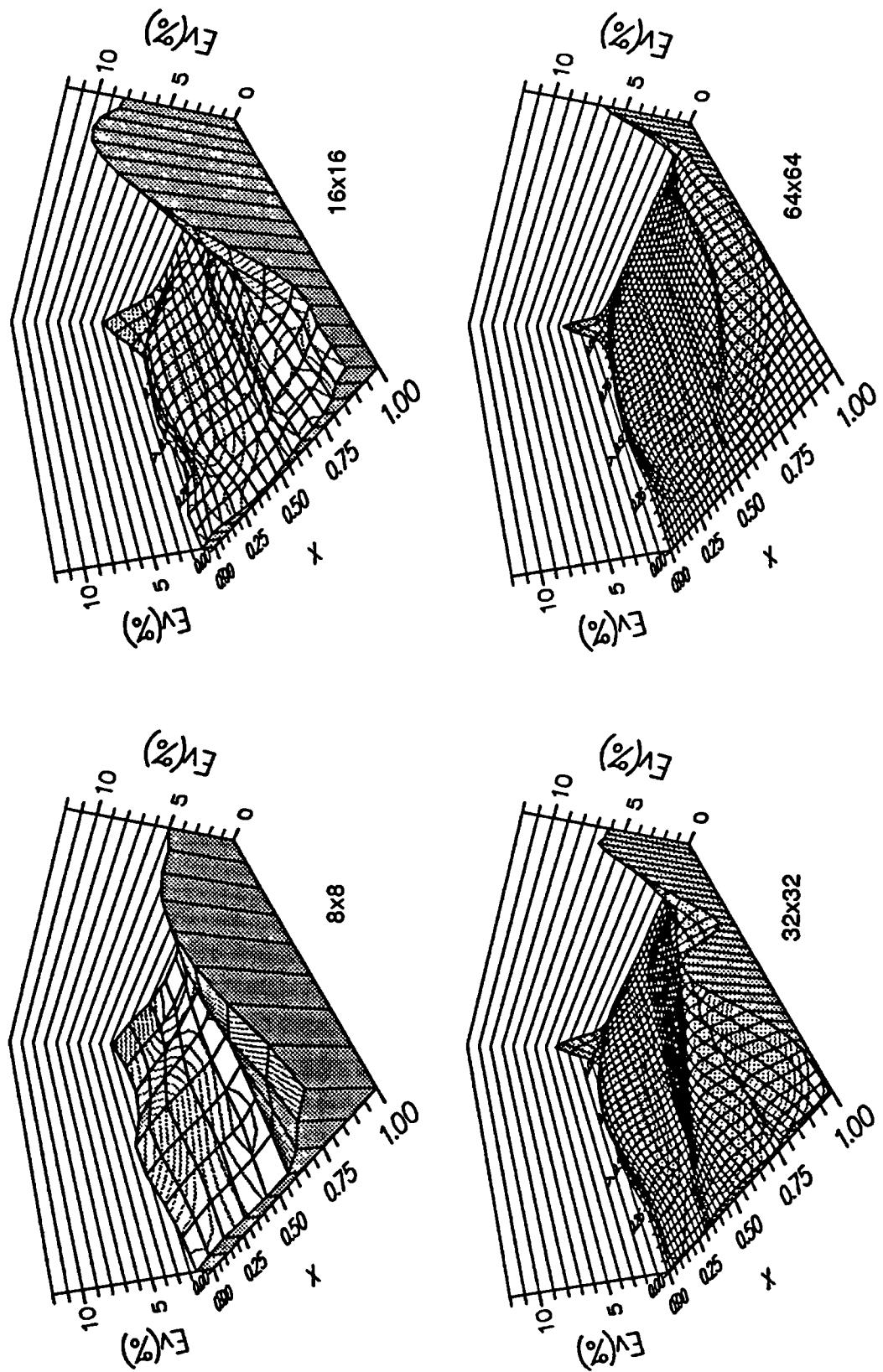


Figure 6.18 (continued)

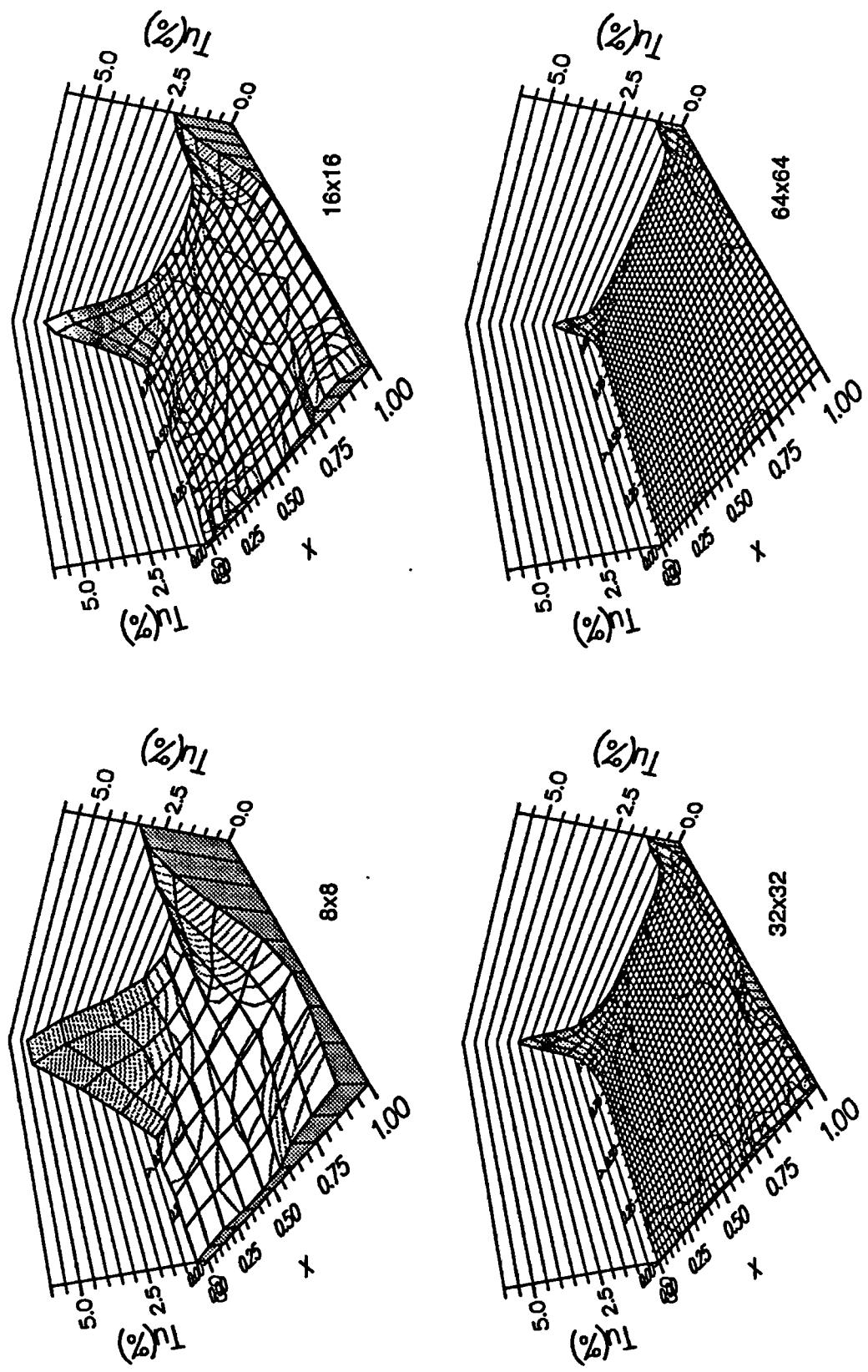


Figure 6.19 Tau error C_1 on different grids for u (T_u) and v (T_v) velocity vector components at $Re=1000$

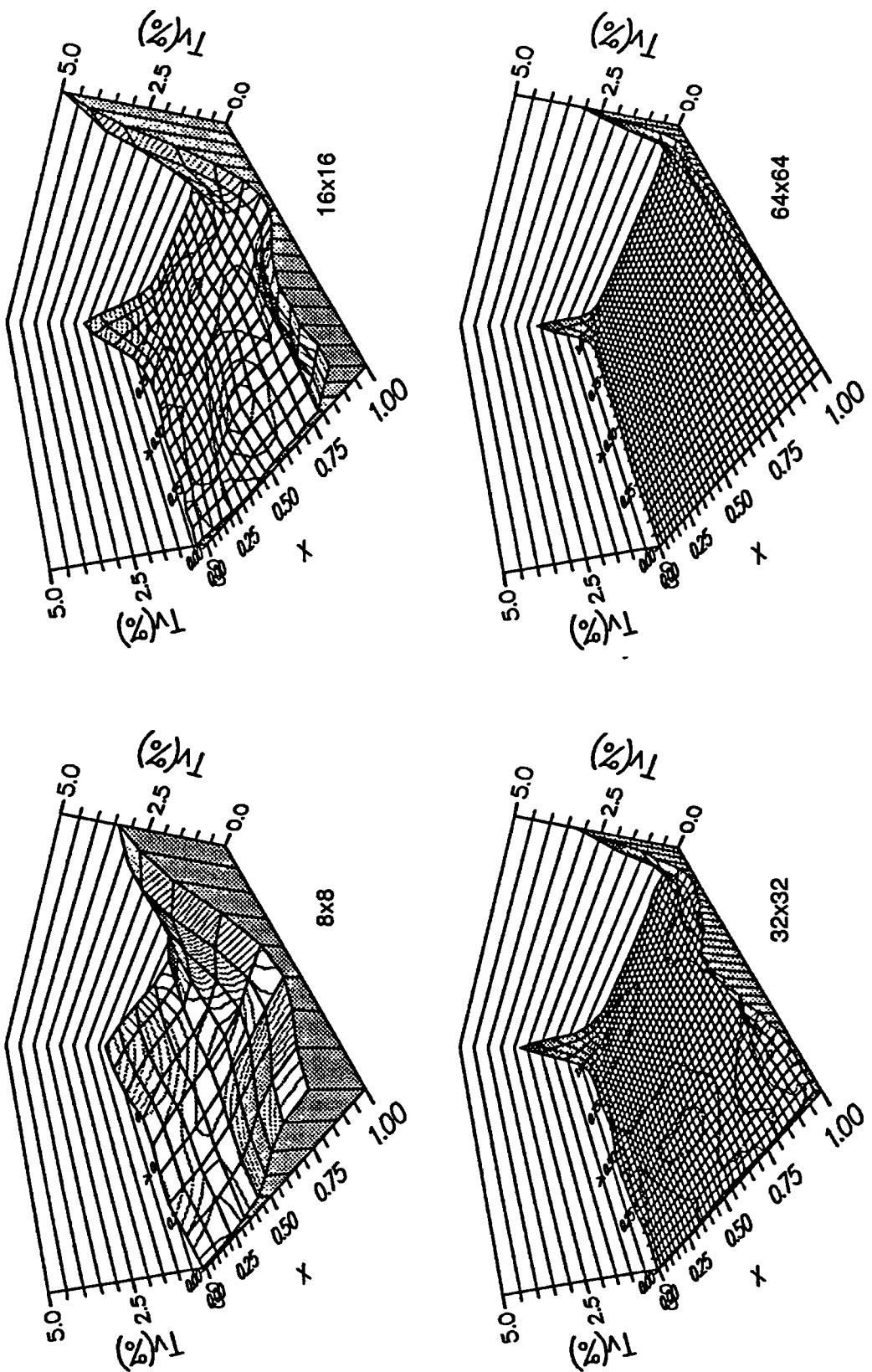


Figure 6.19 (continued)

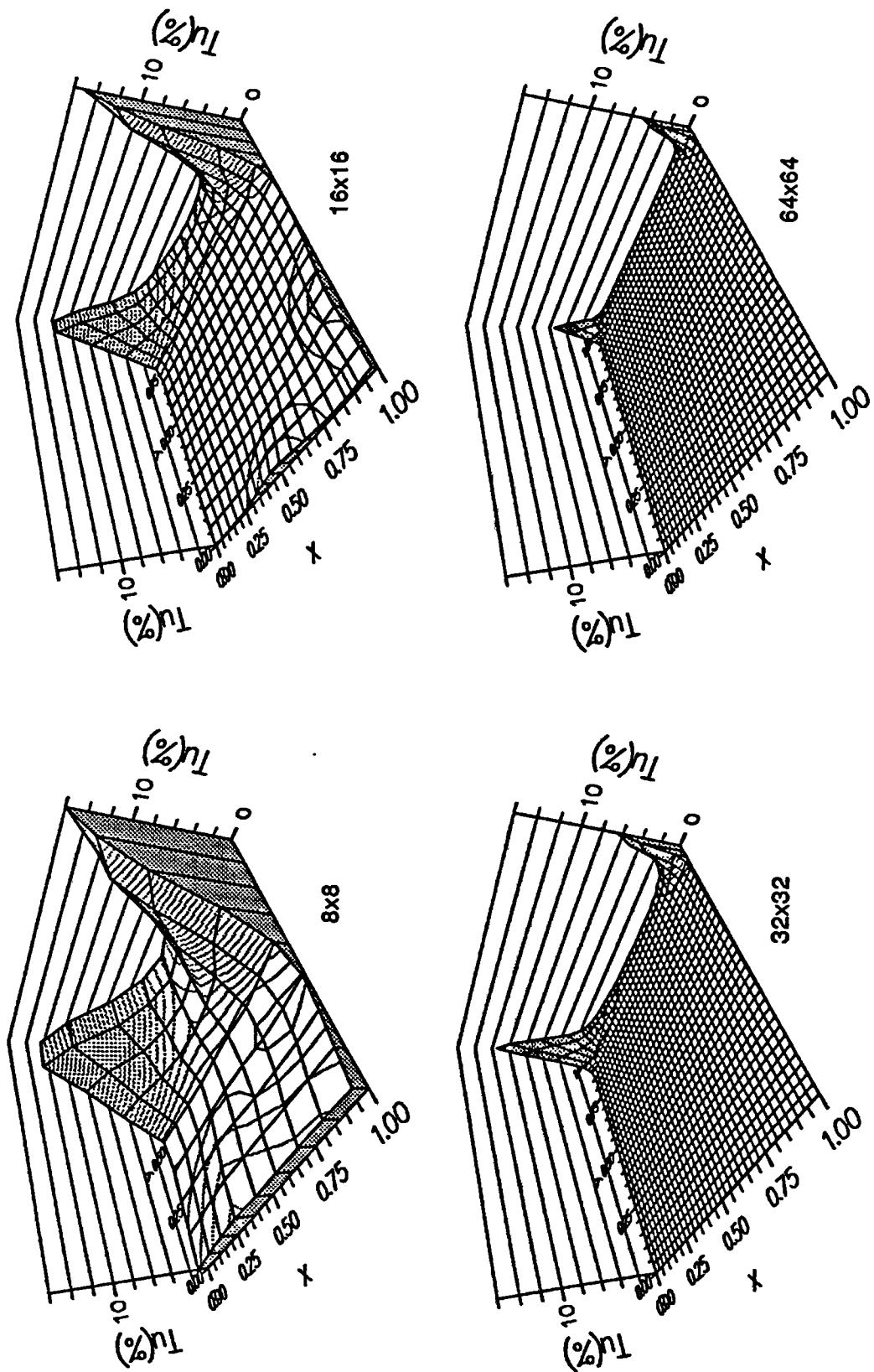


Figure 6.20 Tau error C_2 on different grids for u (T_u) and v (T_v) velocity vector components at $Re=1000$

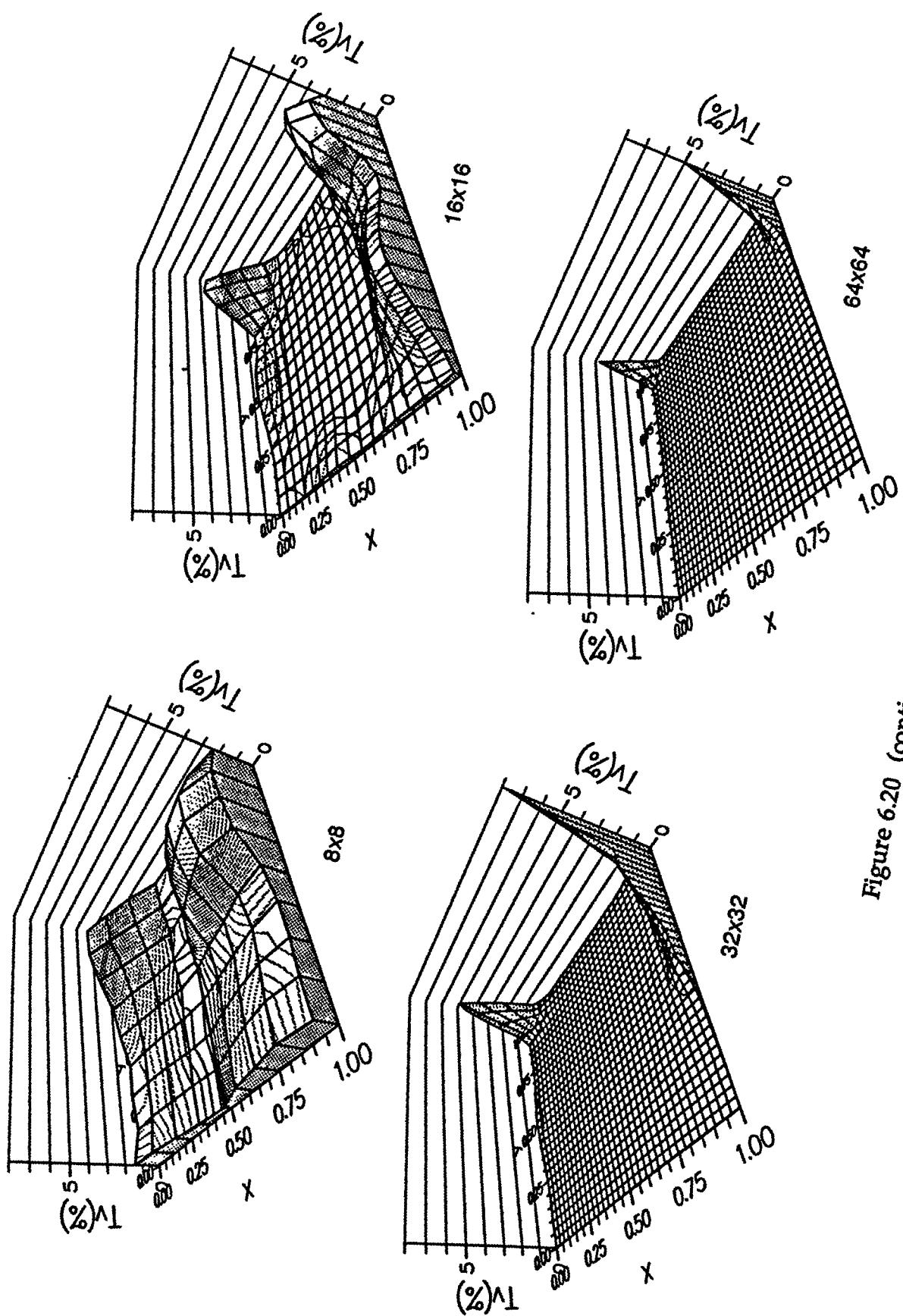


Figure 6.20 (continued)

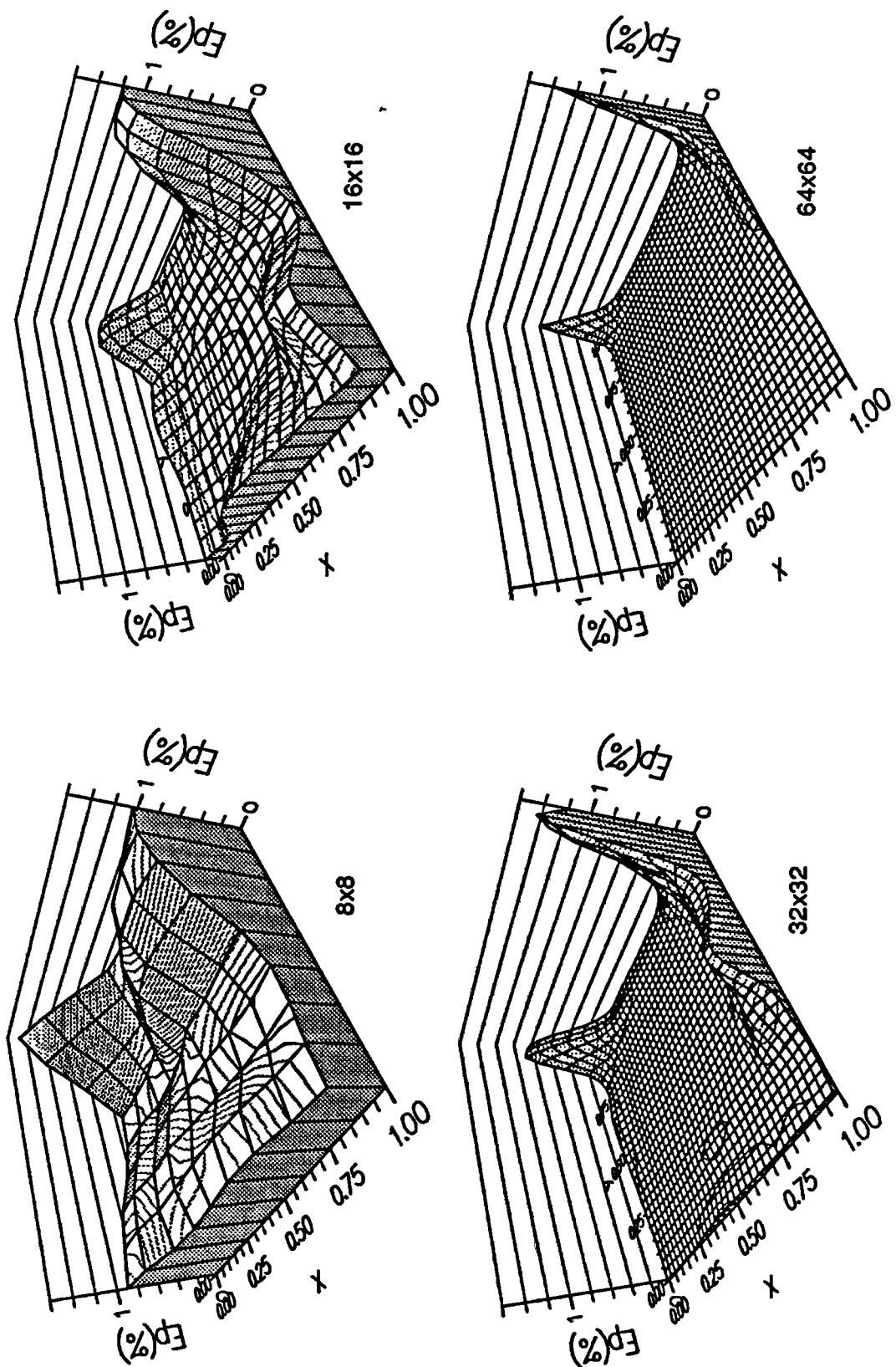


Figure 6.21 Error C2 for pressure (E_p) on different grids at $Re=1000$

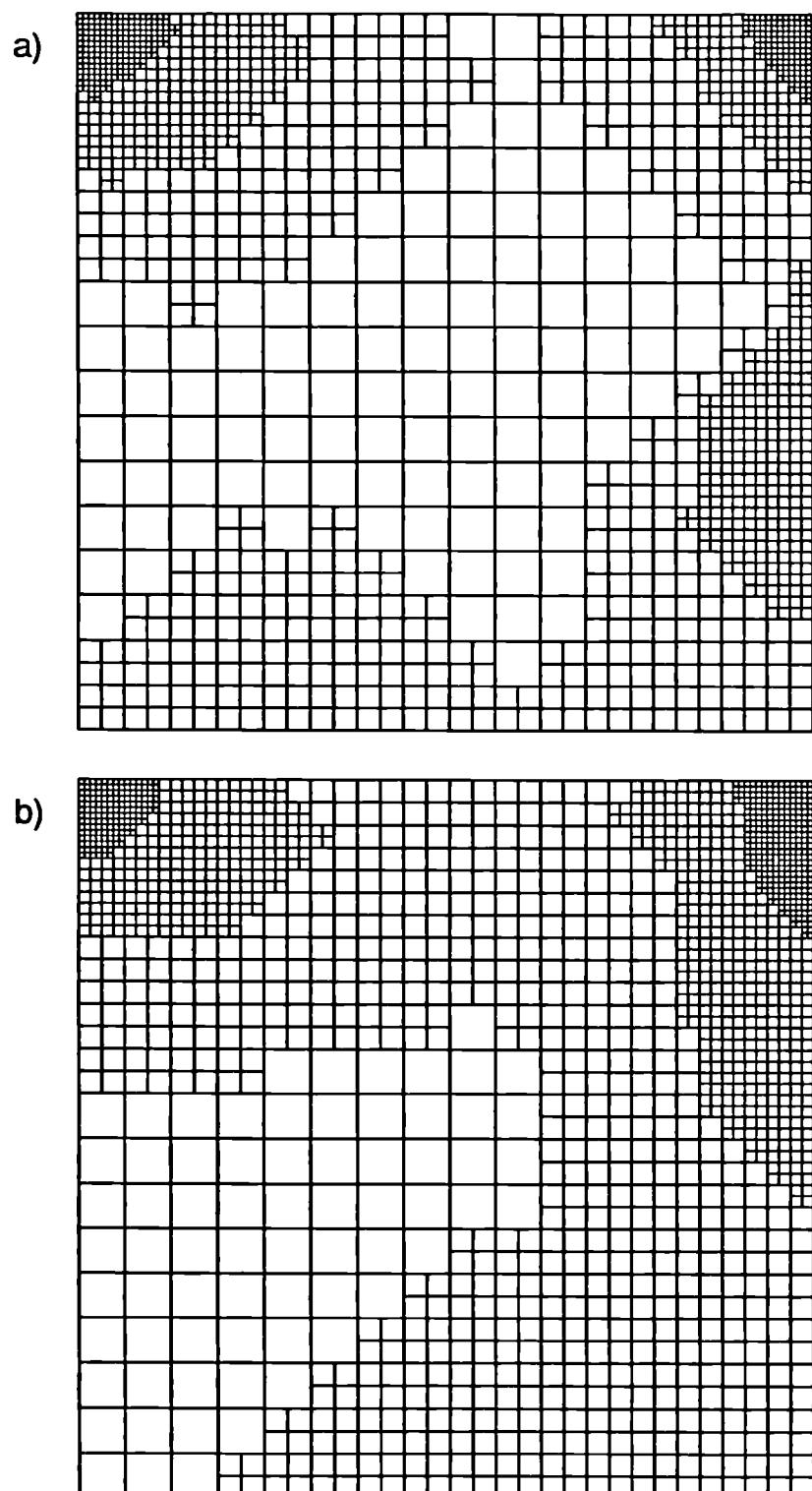


Figure 6.22 Locally refined meshes at $\text{Re}=1000$ based on criterion C1 (1378 CVs) a) and C2 (1645 CVs) b).

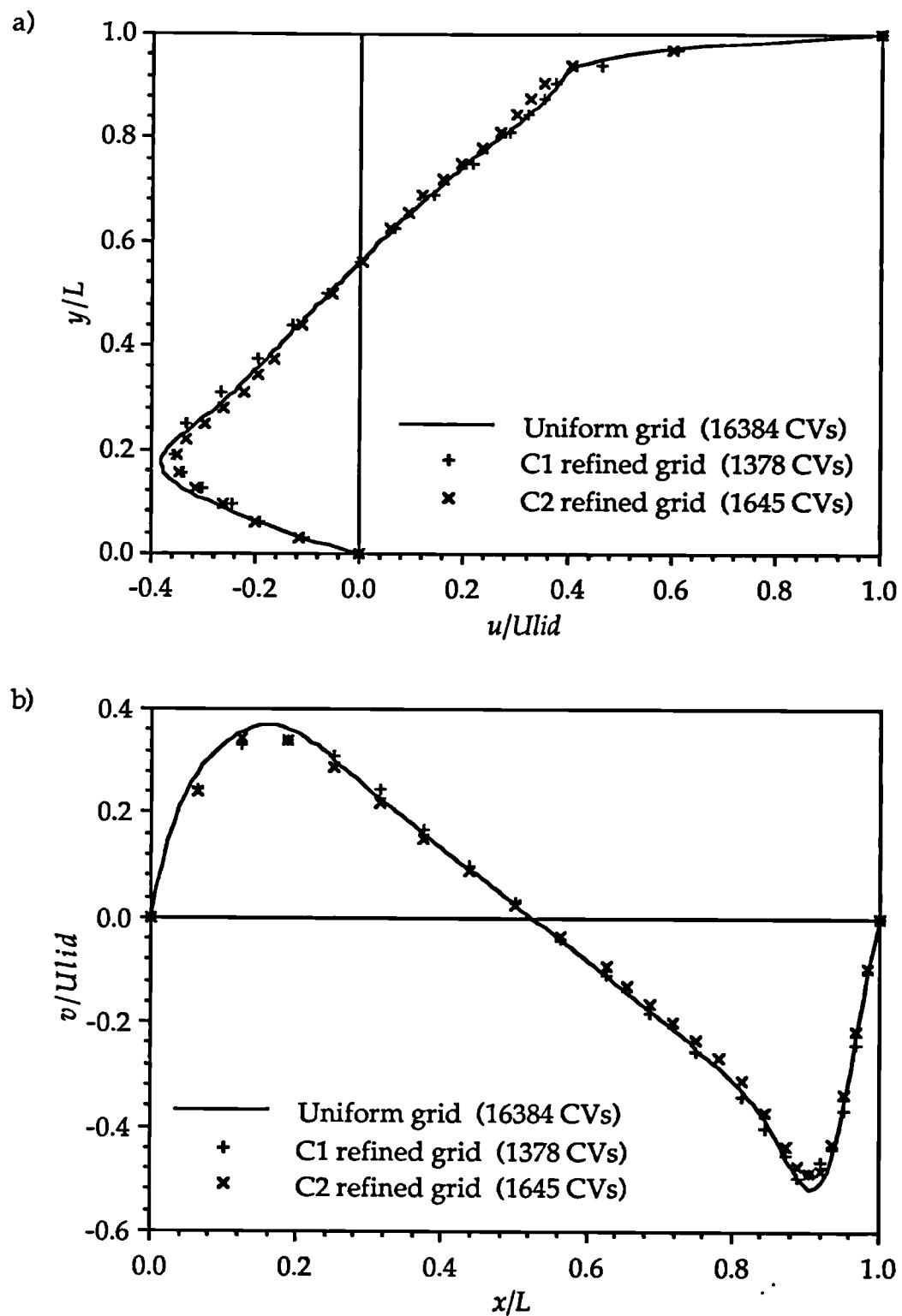


Figure 6.23 Velocity profiles along vertical a) and horizontal b) line obtained on uniform and locally refined grids for $Re=1000$.

Method	CVs	Iterations	CPU
Re=100			
SGM	4096	552	755.70
FMG	4096	32	52.62
FMG-LRC1	916	32	15.74
FMG-LRC2	712	32	12.93
Re=1000			
SGM	16384	998	20976.73
FMG	16384	31	658.53
FMG-LRC1	1378	28	65.75
FMG-LRC2	1645	29	79.06

Table 6.4 Number of control volumes, fine grid iterations and computing time required for different methods to achieve convergence.

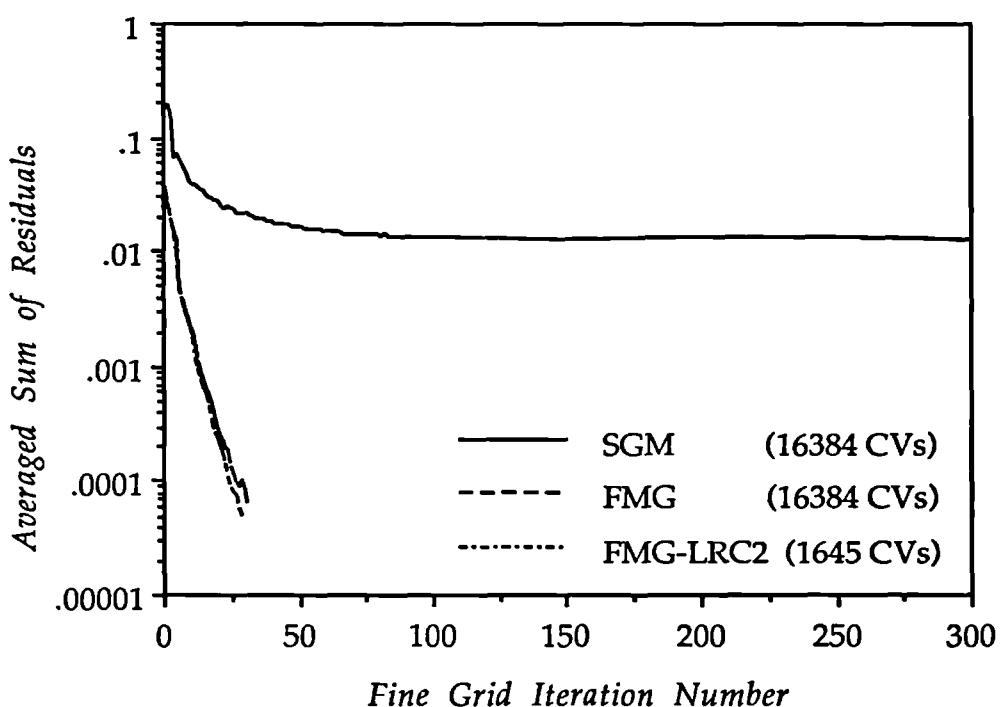


Figure 6.24 Single grid, multigrid and multigrid with local grid refinement convergence histories at Re=1000.

better than those obtained with 64×64 grid. The maximum v velocity vector component is predicted to within 2.7% with respect to the velocity of the driving lid. Tightening the criteria T^* on which the refinement is based would reduce these differences further, but of course, the number of computational points would increase as well.

The savings in number of computational points are larger than those presented by Smith [1990], where similar accuracy was achieved using 5.23 times more control volumes than in the present study. The reason was probably that he had to generate structured locally-refined blocks, extending the regions marked for refinement quite considerably. At the same time his use of the hybrid differencing scheme probably resulted in the need for a rather fine mesh at $Re=1000$ in order to get second order accuracy in the most of the computational domain.

The benefits of using multigrid and local grid refinement, as can be seen from Table 6.4, are great. As can be seen from the graph presented in Figure 6.23, the convergence rate of the multigrid method coupled with the local grid refinement is the same as the multigrid applied to uniformly refined meshes. The savings in CPU time are between 75% for $Re=100$ and 88% for $Re=1000$, compared to the FMG without local refinement. These figures are accompanied by a reduction in the number of control volumes which is 83% for $Re=100$ and 90% for $Re=1000$. It should be noted that the overhead of visiting the coarser grids during multigrid iteration, is not reduced proportionally to the reduction of the number of cells on the finest grid due to the use of local refinement, which is one of reasons why the reduction in the number of control volumes does not produce the same reduction of CPU time.

6.2.3 Influence of Use of Different Linear Solvers within the SIMPLE Algorithm on Multigrid Convergence Rate

In Chapter 4 a number of iterative techniques for the solution of systems of linear algebraic equations were presented. Each of those techniques have different convergence properties, and different numbers of arithmetic operations necessary to perform an inner iteration. These

solvers are an ingredient of the overall iterative schemes of the present CFD algorithms.

	16x16		32x32		64x64		128x128	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
BCG	37	3.75	35	15.09	31	58.89	32	243.1
ICF	37	3.33	35	13.04	32	51.64	32	214.8
GS	37	3.55	35	13.38	32	52.45	32	218.6

Table 6.5 Number of fine grid iterations and CPU time using different solvers for the linearised systems of algebraic equation

The results presented in Table 6.5 for the lid-driven cavity flow at $Re=100$ were obtained using the Incomplete Cholesky preconditioned Bi-Conjugate Gradient (BCG), Incomplete Cholesky Factorisation (ICF) and Gauss-Seidel (GS) iteration methods described in Section 4.2 as smoothers for the discretised momentum equations. The convergence of the inner iteration sequence was monitored and stopped after the initial residual was reduced by an order of magnitude. The UD differencing scheme was used. The Table shows clearly that the overall convergence rate does not depend on the choice of the inner iterative scheme when applied in this fashion. The number of fine grid iterations is the same for all implementations, and the variation in CPU time depends completely on the number of operations necessary to perform one inner iteration. The CPU for BCG implementation is the largest. It is obvious that the specified solver convergence tolerance was not tight enough for the preconditioned BCG to exhibit its good convergence properties. However, as was explained in Section 4.1, there is no need for the solution of the linearised system to be converged to a very tight tolerance. Accordingly, in all further studies reported herein the GS method was used.

6.3 FLOW THROUGH ORIFICE PLATE

From now on attention is turned to more challenging turbulent flow cases. The first is turbulent air flow through a circular pipe containing an orifice plate at Reynolds number 18,400 and $\beta = 0.5$, where β is the orifice diameter ratio (the orifice throat diameter divided by the pipe diameter). Nail [1991] presented extensive measurements obtained by 3-D LDA of the flow upstream and downstream of the orifice plate. The geometry of the pipe and the orifice is presented in Figure 6.25.

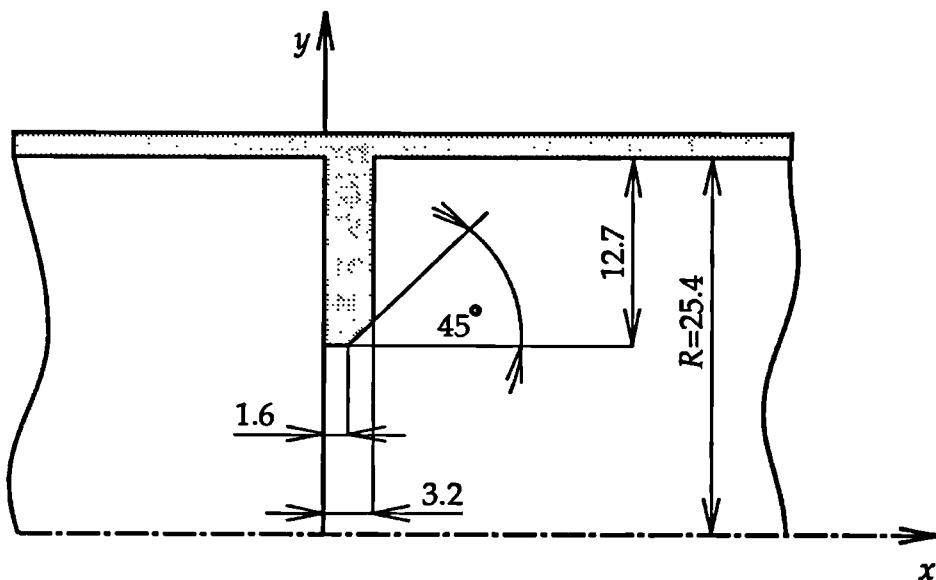


Figure 6.25 Geometry of orifice metering test section

Outlet boundary conditions are imposed seven diameters downstream, and the inlet boundary conditions two diameters upstream of the orifice plate. The reference pressure location selected for the calculation was at the grid point adjacent to the pipe wall just downstream of the inlet plane and the absolute pressure specified there was 134 kPa. The air temperature was set at 300K, and the flowrate at $1.356 \times 10^{-2} \text{ kg/s}$. The kinematic viscosity $\nu = 1.568 \times 10^{-5} \text{ m/s}^2$ and density $\rho = 1.177 \text{ kg/m}^3$ were treated as constant. The measured inlet velocity and turbulence intensity profiles are given in Figure 6.26. The inlet conditions for k were calculated from the measured fluctuating velocity components according to expression (2.30). The turbulence length scale l was specified to be constant and equal to $0.07R$.

over the whole inlet boundary as recommended by Nail [1991]. The inlet value of the turbulent kinetic-energy dissipation rate was then evaluated using the following expression (Launder & Spalding [1974]):

$$\varepsilon = \frac{C_\mu^{0.75} k^{3/2}}{l} \quad (6.2)$$

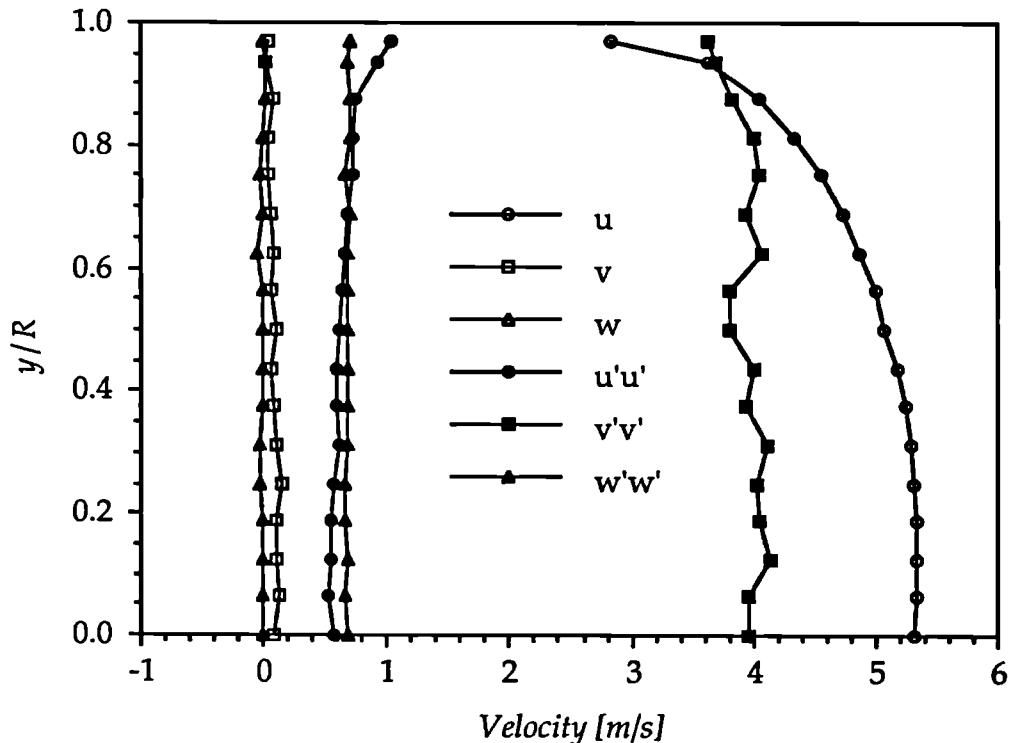
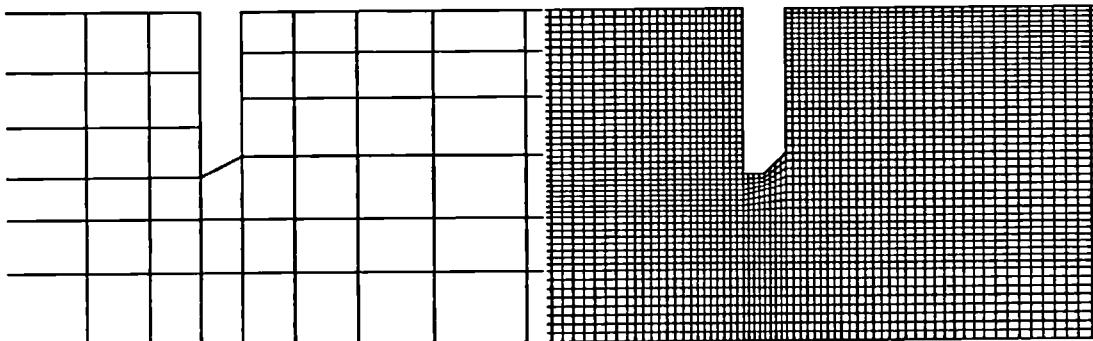


Figure 6.26 Measured inlet velocity profiles

The coarsest (156 CVs) uniform grid and details around the orifice plate of the finest (9792 CVs) such grid used in the calculations are shown in Figure 6.27. Further uniform refinement was not possible since it produces too many grid nodes within the viscous sublayer where the log-law is no longer valid. The streamline pattern obtained using CD scheme and the 9792 CVs mesh is shown in Figure 6.28. Comparisons between the measured and calculated mean velocity vector field are shown in Figure 6.29.



a) The coarsest numerical mesh



b) Detail around the orifice plate for the coarsest (left) and the finest (right) meshes

Figure 6.27 Coarsest and finest uniform grids used in calculation

In general, the predictions are in close agreement with the measurements. The calculation shows that the main flow reattaches to the pipe wall at $x/R=4.44$, while the measured value is 4.5. The calculated velocity distribution follows the measured one well in the upstream section of the pipe. One diameter after the orifice, the jet generated by it begins to spread. Further downstream, a fully-developed turbulent pipe flow is re-established. From the Figure 6.29 it can be seen that the predicted spreading rate of the jet is larger than the measured one. This is more evident from Figure 6.30, where the distributions of predicted and measured u velocity vector component along the centre line are presented. The results obtained are in much closer agreement with experimental data, than the predictions of Nail [1991], probably because of the higher mesh resolution and second-order differencing scheme which were used in the present study.

The variation of wall pressure, presented in Figure 6.31, is another significant measure of the quality of the numerical results. Here again, the

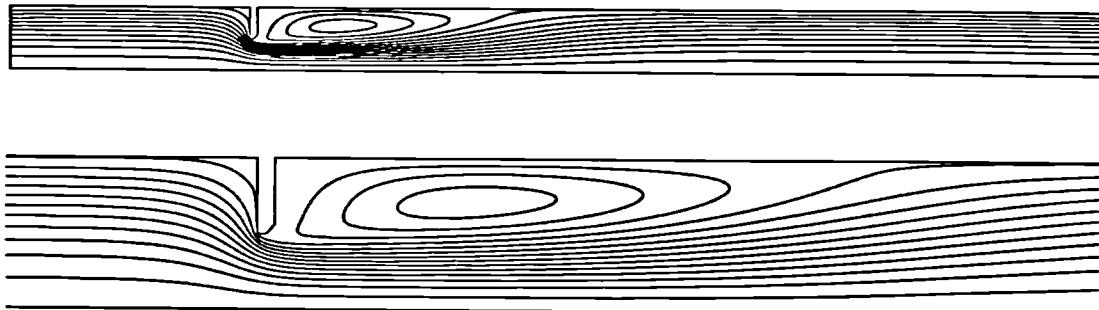


Figure 6.28 Streamlines for flow through orifice plate

profiles upstream and far downstream of the orifice plate are in good agreement with the experimental ones. However, in the region immediately after the orifice, the calculated wall pressure rises faster than the experiment shows.

The adaptive local grid refinement was based on C2 estimation of the tau error. The estimated tau error in velocity was normalised by $v_{ref}=27$ m/s and the kinetic energy of turbulence and its dissipation by $k_{ref}=40$ m²/s² and $\varepsilon_{ref}=2.5 \times 10^5$ m²/s³, respectively. The fields of these error estimations for $-4 \leq x/R \leq 8$ are plotted in Figures 6.32-6.34. The largest errors for all transport variables occur, as one would expect, in the region around the orifice, where the large gradients of all variables exist. The estimated error for the u velocity vector component (Figure 6.32) is large in recirculating regions near the wall, as well. Unfortunately, because of the limitations of application of wall functions, it was not possible to keep refining in this region till the estimated error was below the prescribed value of 1%. The locally-refined mesh based on those fields is shown in Figure 6.35. The patch of the finest mesh starts just upstream of the orifice plate and extends about six diameters downstream, constantly shrinking towards the axis of symmetry. As can be seen from Figure 6.30, this refinement pattern coincides with large spatial variations of the axial and radial velocity components near the axis of symmetry.

As was already mentioned, the uniform grid refinement was limited to the 9792 CVs mesh because of the use of wall functions in near-wall region. However an even finer mesh was created by uniformly refining the cells of this mesh, unless they were in regions where the log-law was no longer valid. The resulting mesh has 36956 CVs and its finest cells correspond to the finest cells of 19308 CVs locally-refined mesh. The velocity and pressure profiles obtained on these two meshes (Figures 6.30 and 6.31) are very similar (the maximum difference in the axial velocity is less than 4% of the maximum velocity), while the reduction in number of control volumes for local refinement is about 48%.

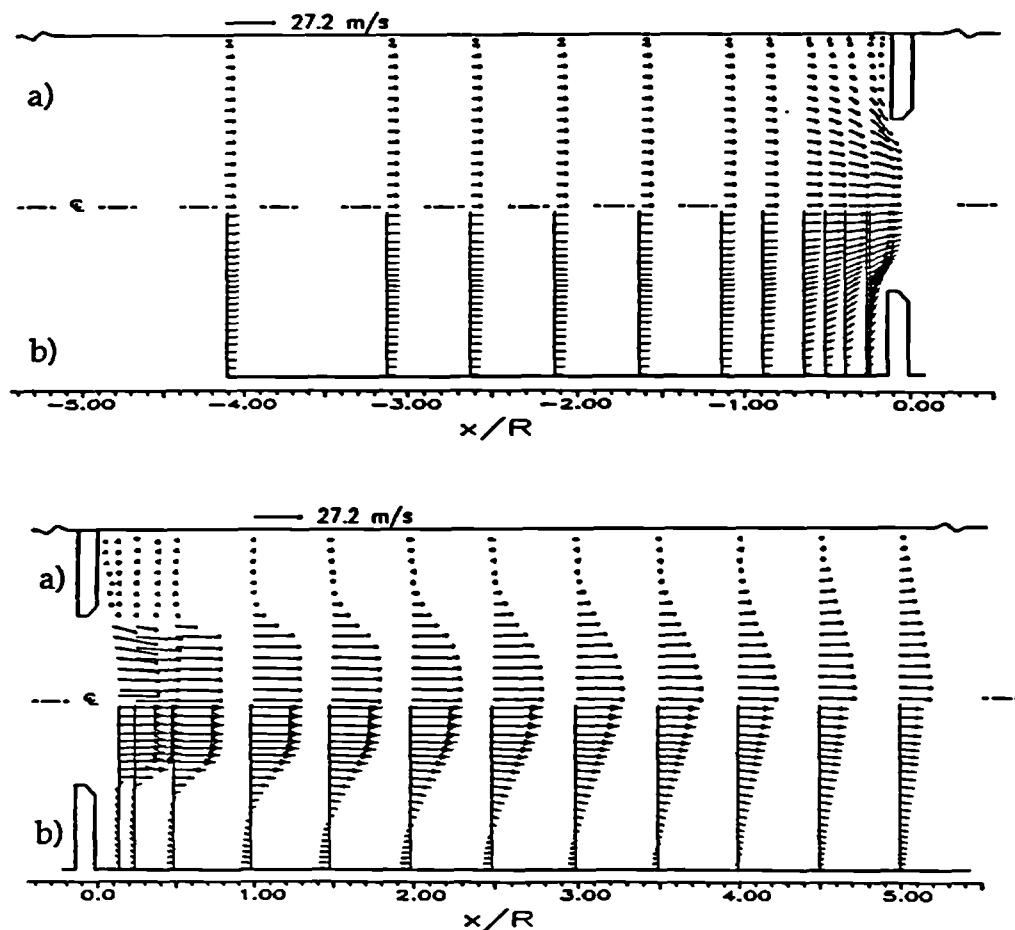


Figure 6.29 Measured a) and calculated b) velocity vector fields for flow through orifice plate ($x/R = -4$ to 5)

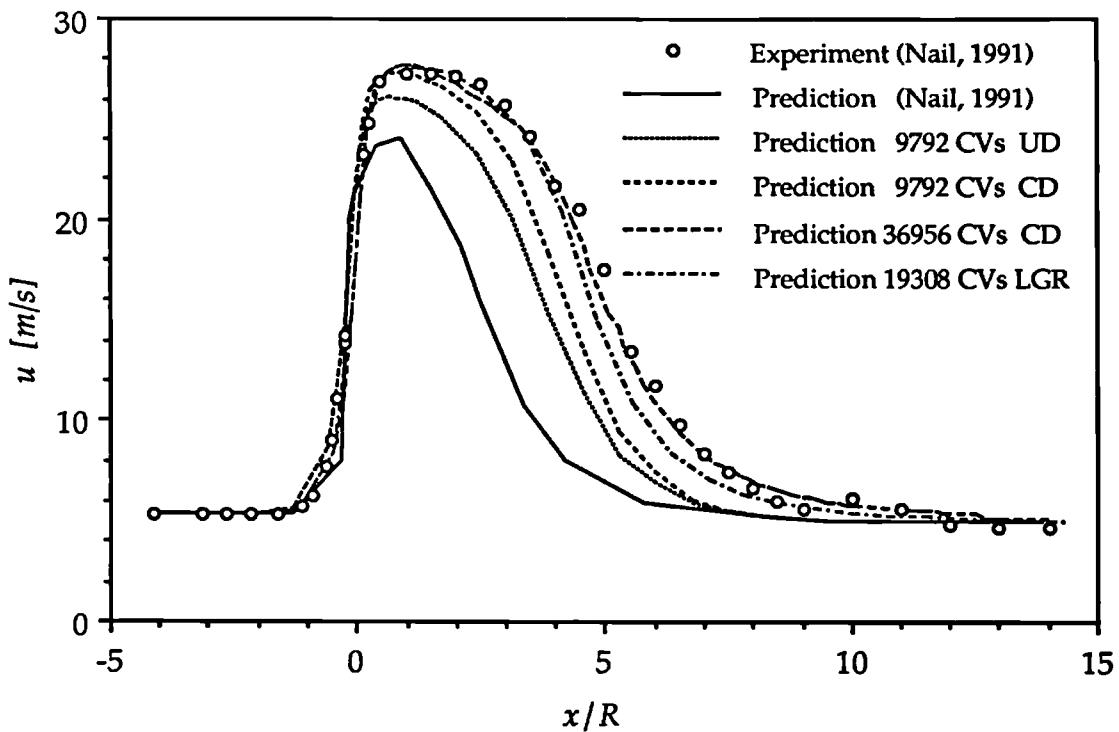


Figure 6.30 Mean axial velocity along symmetry axes line for flow through orifice plate

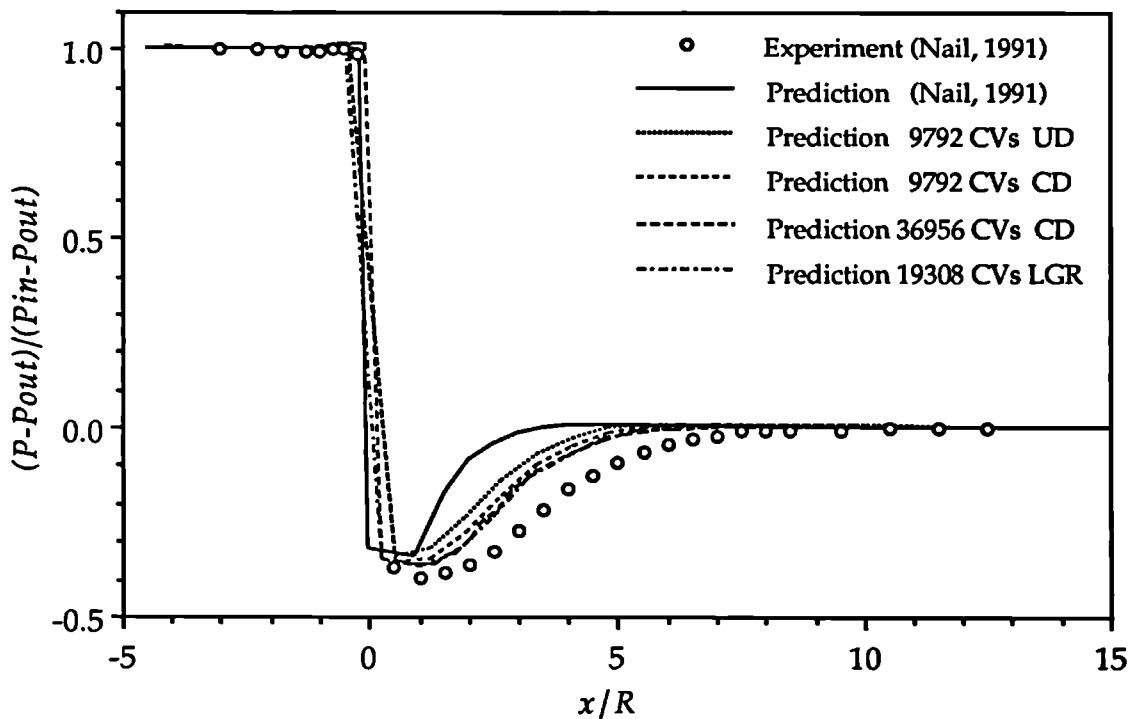


Figure 6.31 Wall pressure distribution

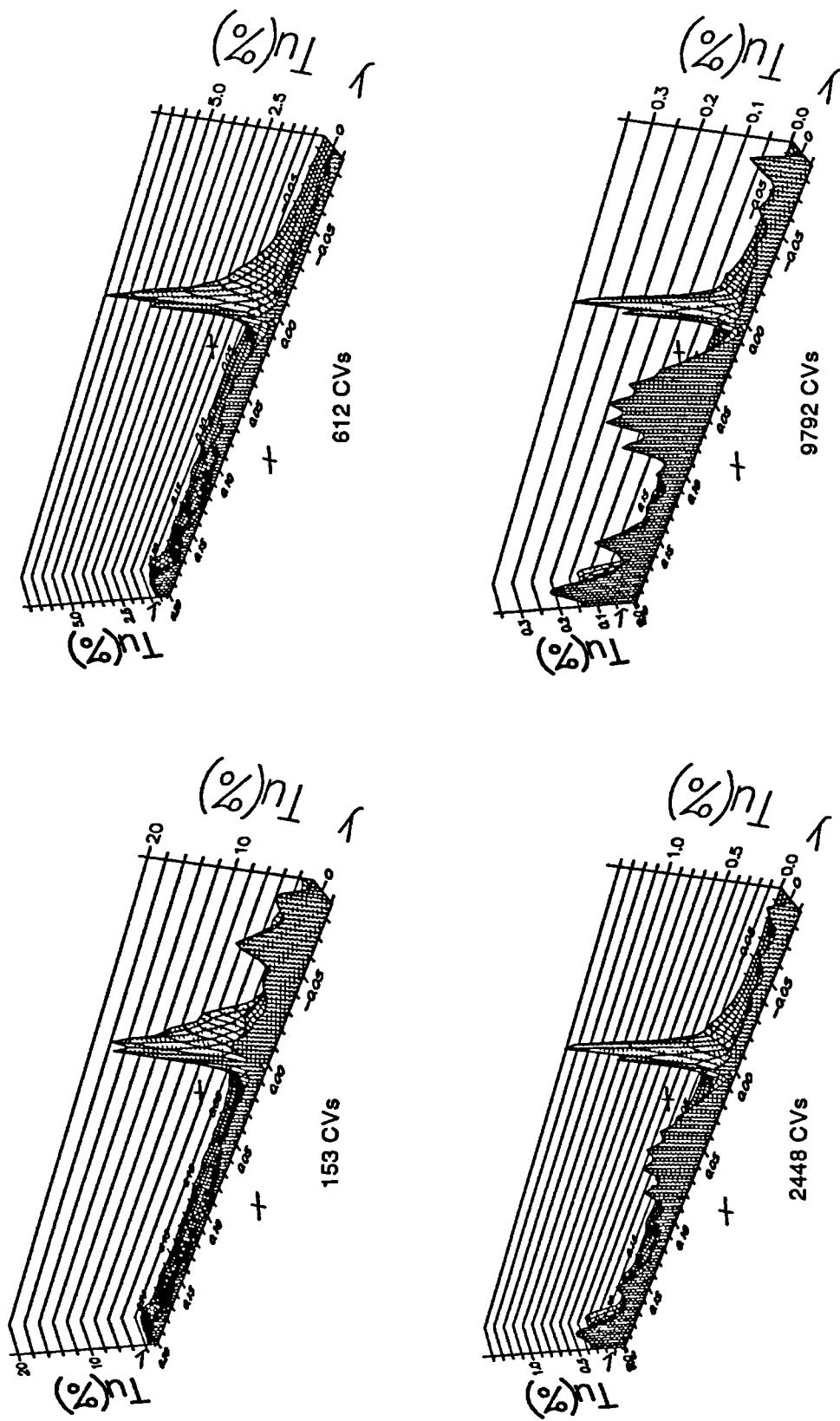


Figure 6.32 Tau error C_2 on different grids for u (Tu) and v (Tv) velocity vector components

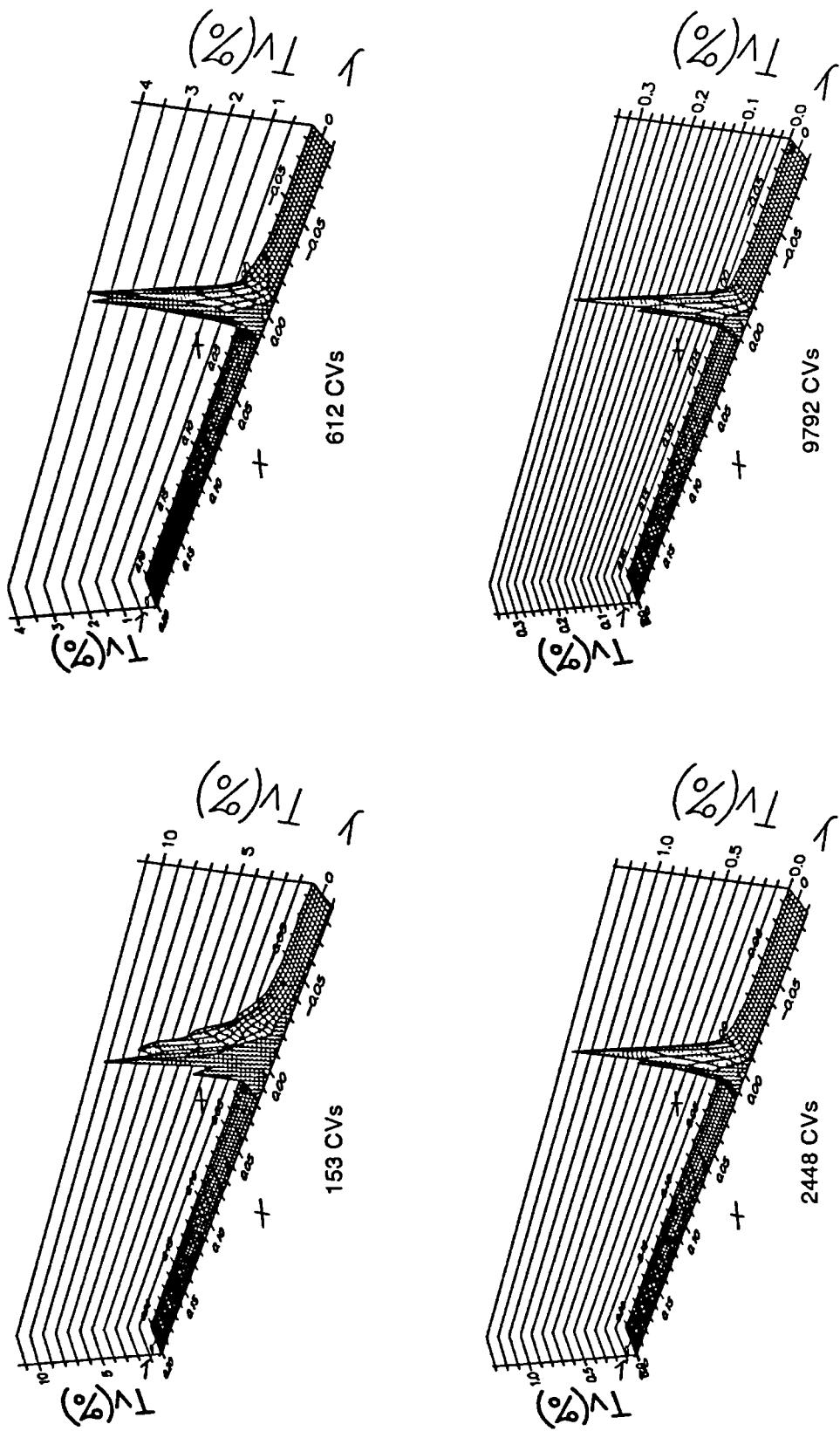


Figure 6.32 (continued)

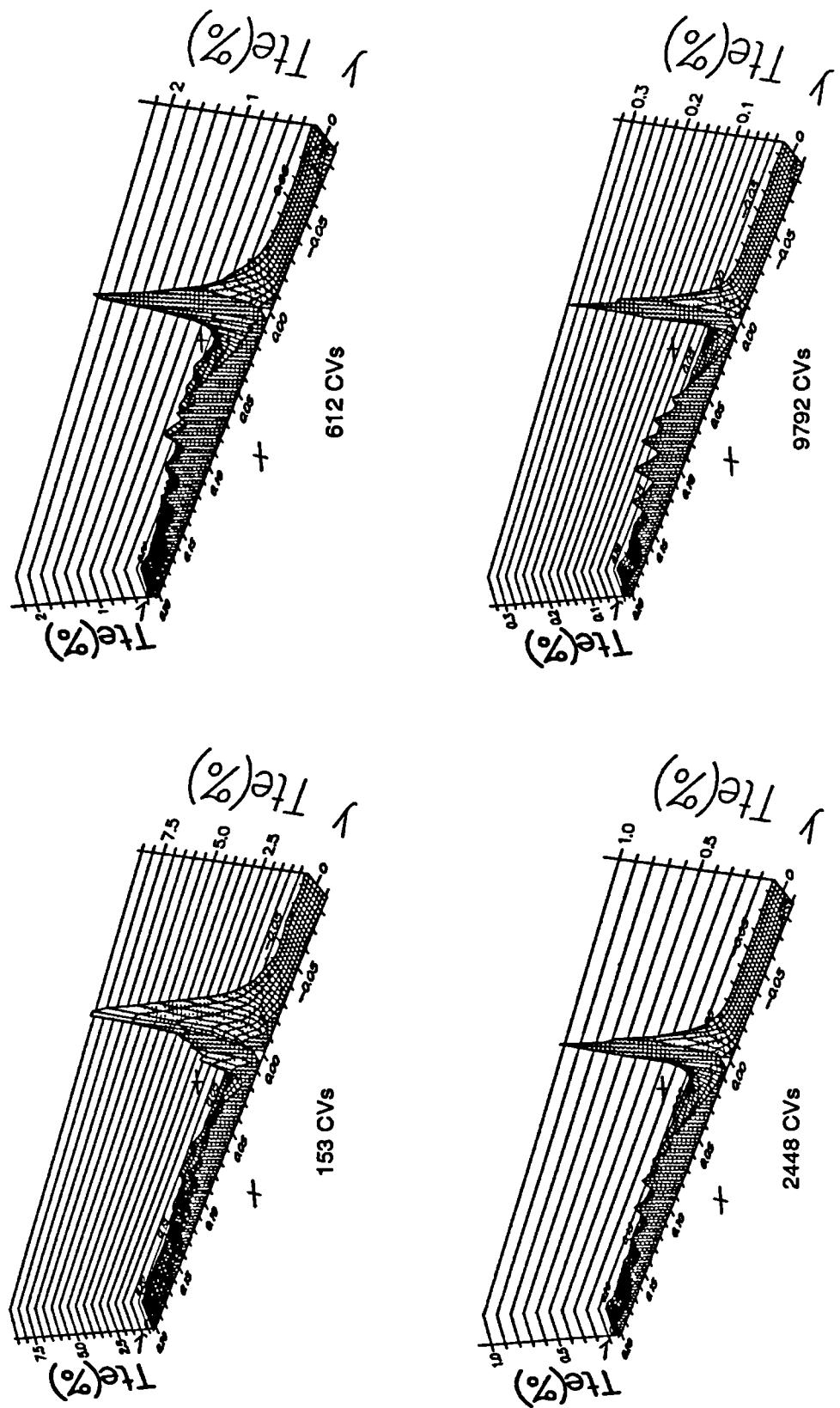


Figure 6.33 Tau error C2 on different grids for kinetic energy of turbulence (T_{ke}) and its dissipation (T_{de})

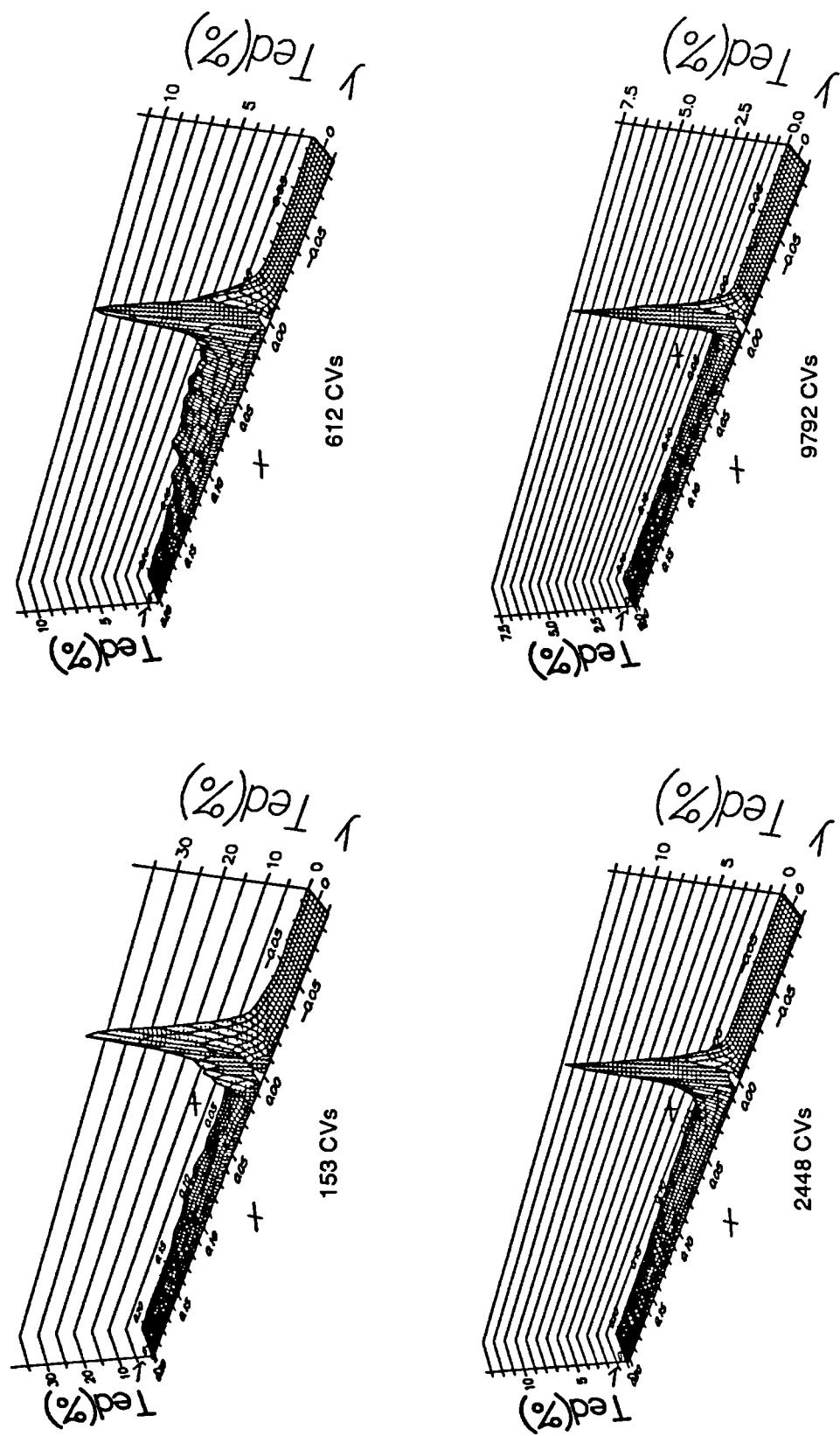


Figure 6.33 (continued)

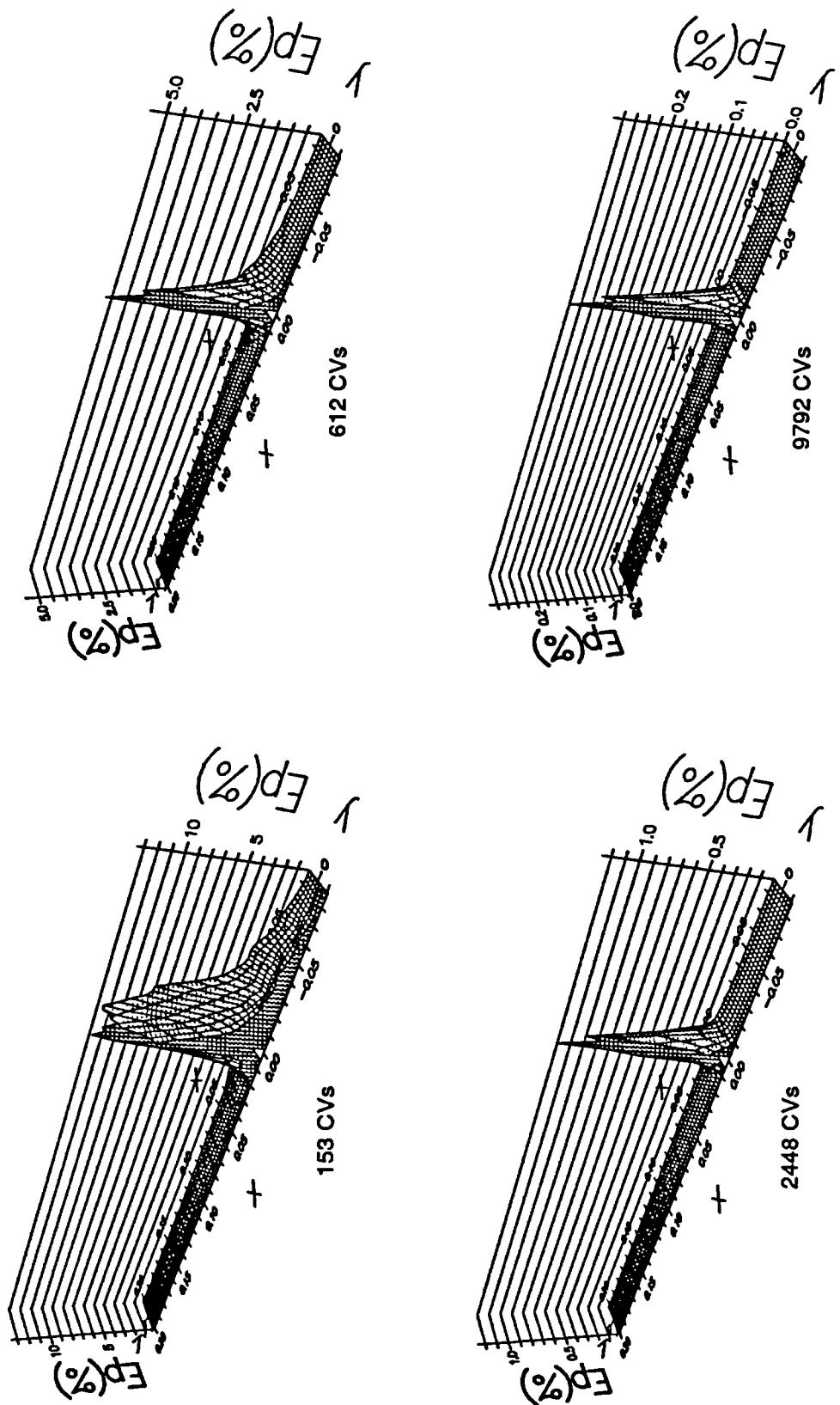


Figure 6.34 Error C_2 for pressure (E_p) on different grids

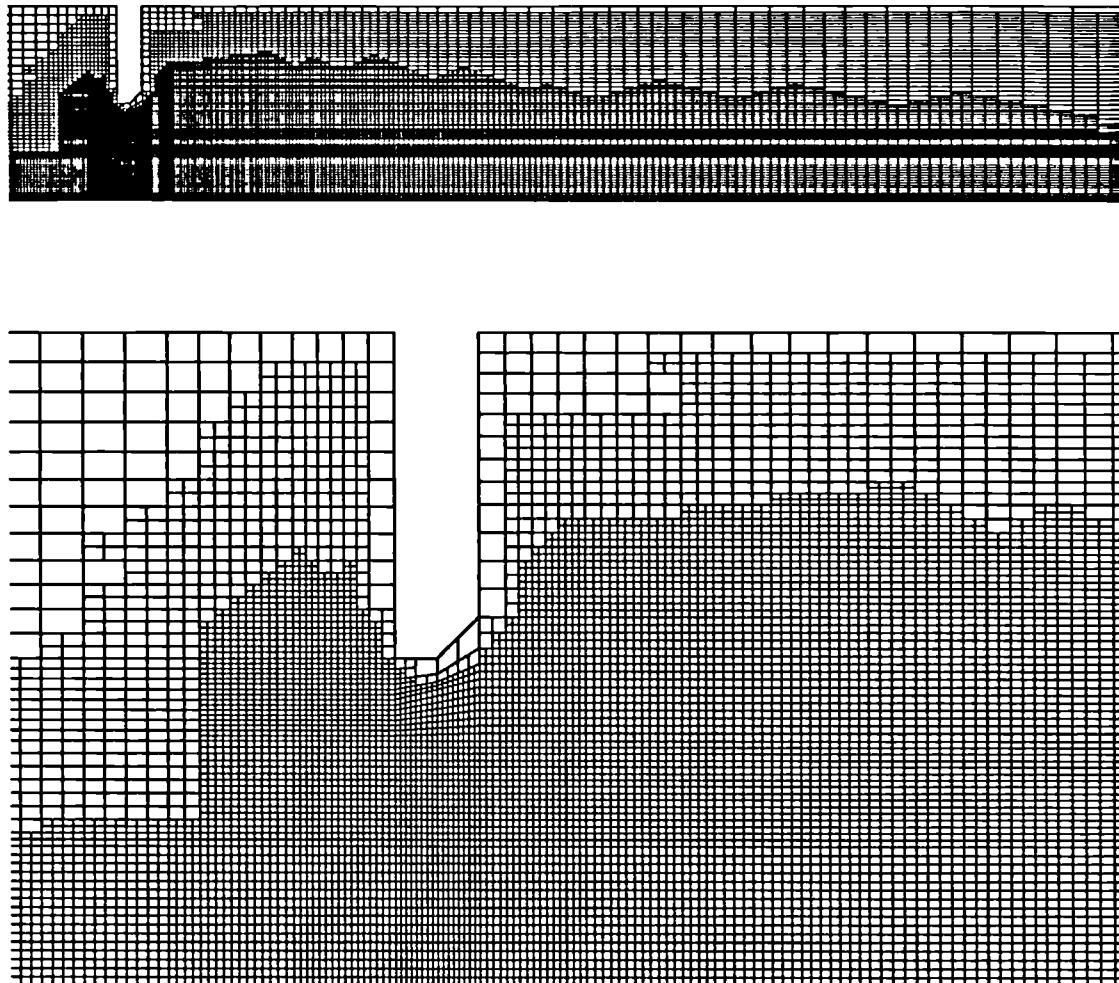


Figure 6.35 Locally refined mesh - 19308 CVs

The performances of the SGM and FMG methods are summarised in Table 6.6, which compares the number of fine grid iterations and CPU times. In order to obtain MG convergence on the 36956 grid the coarsest grid could not be used for multigrid. A possible reason is that the representation of the problem on the coarsest grid was so poor that the corrections extrapolated from this grid were not only of little use, but hindered the convergence.

From Table 6.6 it is also possible to see a large increase in the number of fine grid iterations as the FMG moves to next finer grid. Moving from the

CVs	612		2448		9792		36956		19308 lgr	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
Central Differencing Scheme										
SGM	95	47.07	214	432.2	733	6760	2281	65293	960	20787
FMG	105	55.51	85	216.5	122	1263	243	9548	152	4997
FMGU	113	71.07	73	206.9	79	1061	141	7742	126	4108
SGM/FMGU CPU										
	0.66		2.10		6.37		8.43		5.06	
Upwind Differencing Scheme										
SGM	65	26.42	208	361.5	723	6040	2099	60309		
FMG	86	53.09	64	209.0	95	1254	120	6631		
SGM/FMG CPU										
	0.69		1.74		4.97		9.15			

Table 6.6 Fine grid iteration and computing time required to obtain convergence for flow through the orifice plate at $Re=18400$.

9792 CVs grid to 36956 CVs grid, the number of fine-grid iterations increases from 122 to 243. This large increase is only noted for the CD scheme. Because of this the practice was tried of using the UD scheme on the coarser grids and the CD scheme only on the finest mesh. Of course, the final solution on the finest grid is still second-order accurate. This strategy was successful, reducing the number of fine grid iterations on 36956 CVs to 141 (see Table 6.6 row marked as FMGU). The same strategy has also been tried for the driven-cavity cases in the expectation of seeing some further improvement for them, but in fact the number of fine-grid iterations increased by between 10% and 20% compared to the use of the CD differencing scheme on all meshes. The probable explanation for this is that in the cavity calculations, the cell Péclet numbers are relatively small, allowing use of the pure CD scheme even on the coarsest mesh. However, in some regions of the orifice flow, where the Péclet number is large, instabilities and oscillatory fields of corrections could be generated if the CD scheme is used on coarser meshes. This could have two undesired effects, both which can influence the MG performance. Firstly, the convergence of the coarse grid equations might be slowed down, because of

the instability introduced using the higher-order differencing scheme. Secondly, the oscillatory corrections, when extrapolated to the finer grid, could introduce high-frequency error components, and additional fine-grid iterations would be necessary later to reduce them.

The convergence acceleration obtained from multigrid on the five uniformly refined grids is between 8.43 (for the CD scheme) and 9.15 (for the UD scheme). Adaptive local grid refinement does not have a negative influence on multigrid convergence properties. The number of fine-grid iterations for the FMG method applied to 612 CVs mesh is larger than for the SGM method. This supports the conclusion from the above, that the coarsest grid is too coarse to represent the problem from the finer meshes, and corrections from this grid are misleading, increasing the total number of fine grid iterations.

The number of fine grid iterations for the FMG method increases slowly with the increase in number of computational points. For the UD scheme, as can be calculated from Table 6.6, the increase has an exponent of about 0.2. For the CD scheme when comparing the fine grid iterations on the grids 2448 CVs and 9792 CVs this exponent is 0.06 , which is almost optimum performance. However, as the mesh resolution is increased from 9792 CVs to 36956 CVs the number of fine grid iterations increases dramatically from 79 to 141. An explanation for this behaviour was not found.

The increase in number of fine grid iterations with the increase in the number of computational points, although generally not large, is not in accordance with the optimum behaviour (the number of fine grid iterations remains the same) which was observed in the laminar driven cavity flow calculations. The possible reasons for this will be discussed in the final section of this chapter.

6.4 THREE-DIMENSIONAL TURBULENT FLOW OVER CUBICAL OBSTACLE

Predictions of turbulent flow around a surface-mounted cubical obstacle placed in a channel are the subject of this section. Although the geometry is relatively simple, the flow around sharp-edged obstacles contributes directly to the understanding of the flow mechanisms of more complex three-dimensional geometries (Larousse et al. [1991]). The three-dimensional turbulent flow generated, exhibits considerable complexity and is therefore suitable for the evaluation of the computational performance of the present methodology. The predicted results are compared with the measurements of Vasilic-Meling [1976].

The geometrical characteristics of the experimental test section are defined by Figure 6.36. They include the height h of the cube, the lengths L_u and L_d of the solution domain upstream and downstream of the front face, the height L_y and the width L_x of the domain. The solution domain in the x - y planes represents only half of the flow field since symmetry was assumed about the y - z plane bisecting the obstacle. The origin of the coordinate system is located at the midpoint of the bottom edge of the front face of the cube. The inlet boundary conditions were deduced from experimental data and listed in Table 6.7. The turbulence length scale l was specified to be uniform and equal to $0.09L_y$ over the whole inlet boundary (Vasilic-Meling [1976]). The inlet value of the dissipation rate was evaluated using expression (6.2).

Both uniformly- and locally-refined numerical meshes were used for the calculations. The coarsest mesh, composed of 336 CVs, and four uniformly-refined meshes, the finest having 172032 CVs are shown in Figure 6.37a and 6.37b. Further uniform grid refinement, which would have produced a mesh of 1376256 CVs, was not possible because of computer memory limitations. However, five levels of adaptive local grid refinement, being based on criterion C2, were performed, and the resulting mesh, consisting of 186452 CVs, is presented in Figure 6.37c. The error

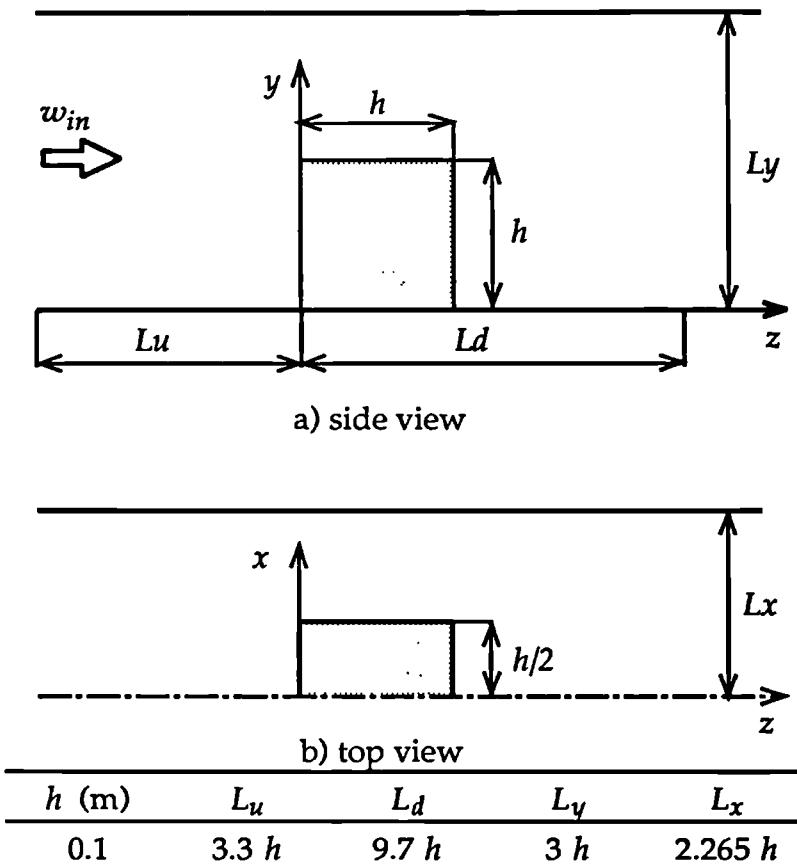


Figure 6.36 Geometry of solution domain shown in y - z and z - x plane

u_{in} (m/s)	v_{in} (m/s)	w_{in} (m/s)	k_{in} (m^2/s^2)
0.0	0.0	25.0	9.38×10^{-2}

Table 6.7 Inlet boundary conditions

estimations were normalised by the values of dependent variables at the inlet boundary (Table 6.7).

The velocity vector and pressure fields on the planes $x=0$ (symmetry plane), $y=h/2$ (horizontal plane passing through the centre of the obstacle) and $z=h/2$ (vertical plane passing through the centre of the obstacle and normal to the symmetry plane) are presented in Figures 6.38, 6.40 and 6.42 respectively. The results presented were obtained on the locally refined

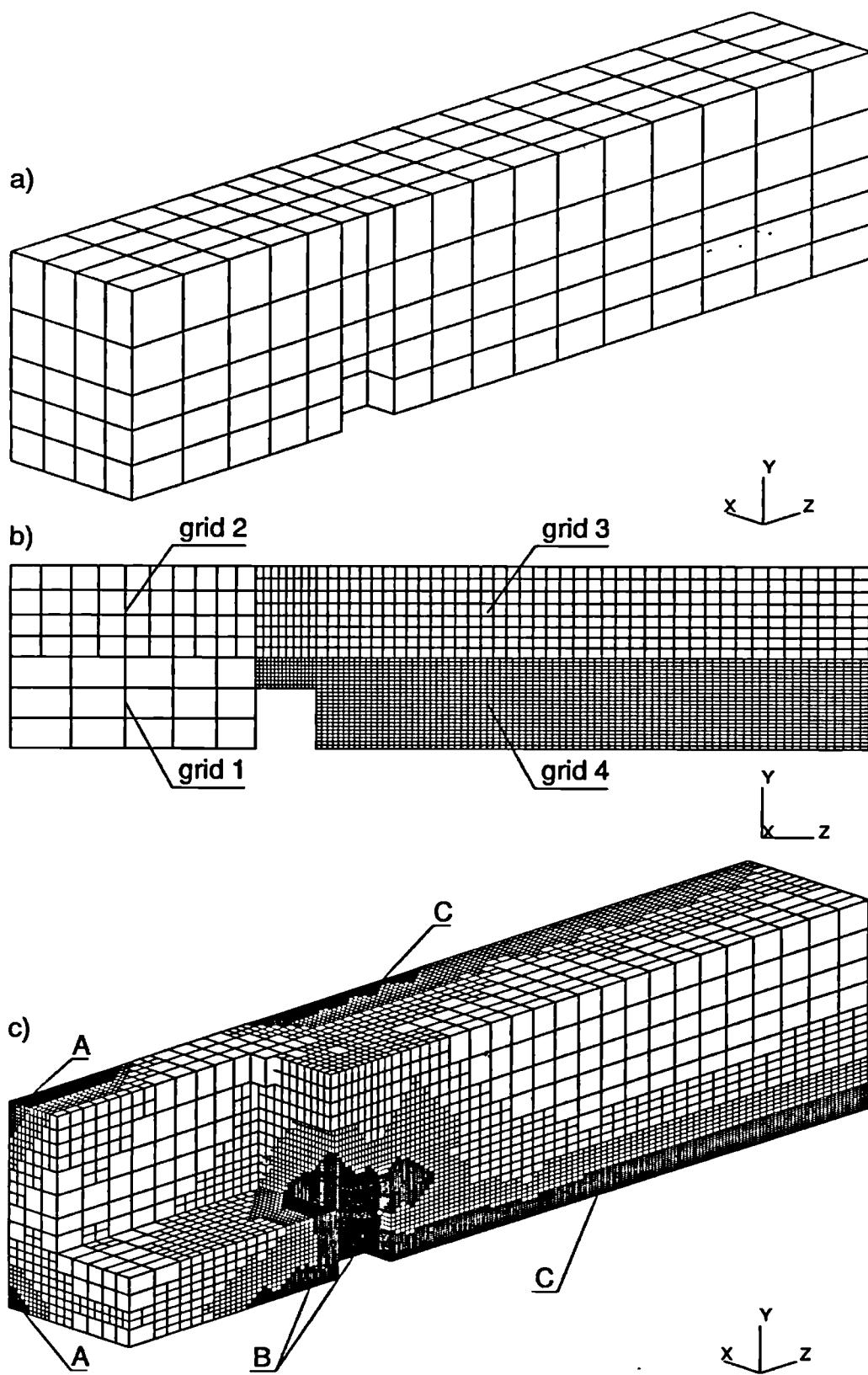


Figure 6.37 Coarsest a), four uniformly b) and locally refined c)
meshes used in calculation

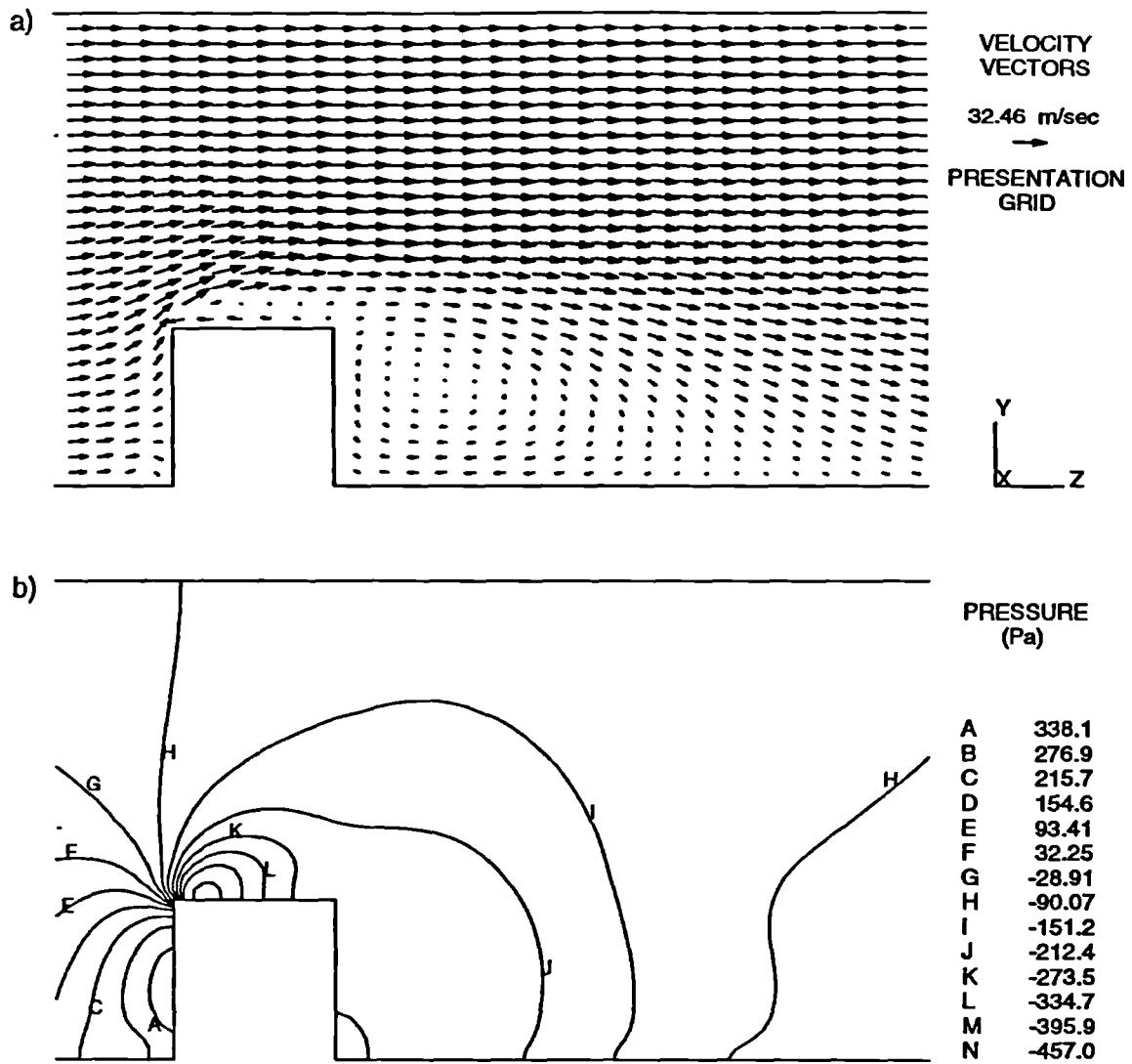


Figure 6.38 Predicted velocity vector and pressure fields in $x=0$ plane on locally refined mesh (186452 CVs)

186452 CVs mesh. In order to better present this data results have been mapped onto a uniform Cartesian mesh. Comparisons between the measured and predicted local pressure coefficient C_p , defined as

$$C_p = \frac{\frac{1}{2} \rho w_{in}^2 - p}{\frac{1}{2} \rho w_{in}^2} \quad (6.3)$$

are given in Figures 6.39, 6.41 and 6.43. The free-stream pressure p_{in} appearing in the above definition was specified at $x=0, y=0, z=-4.5h$.

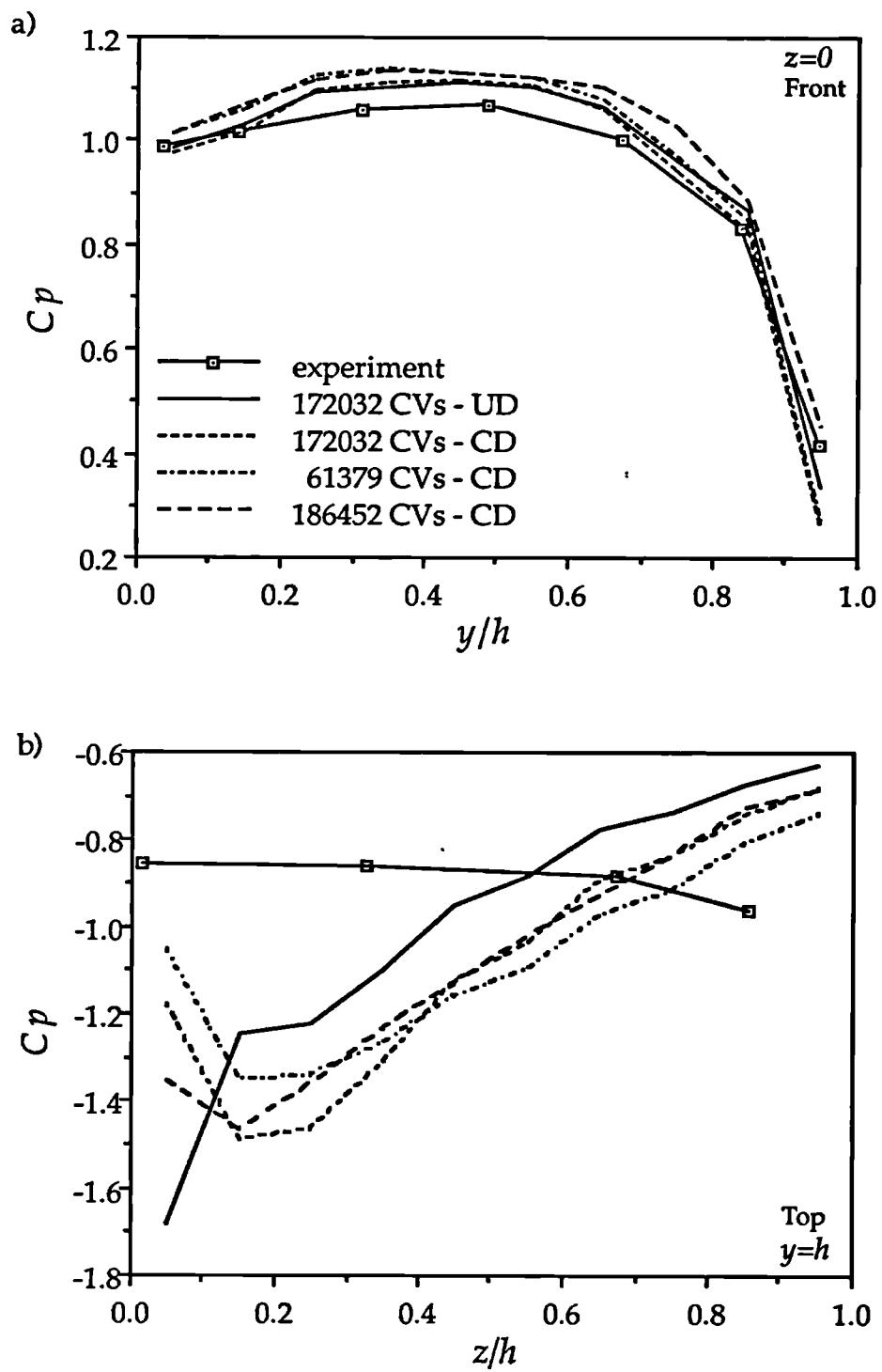


Figure 6.39 Pressure variation around obstacle in plane $x=0$

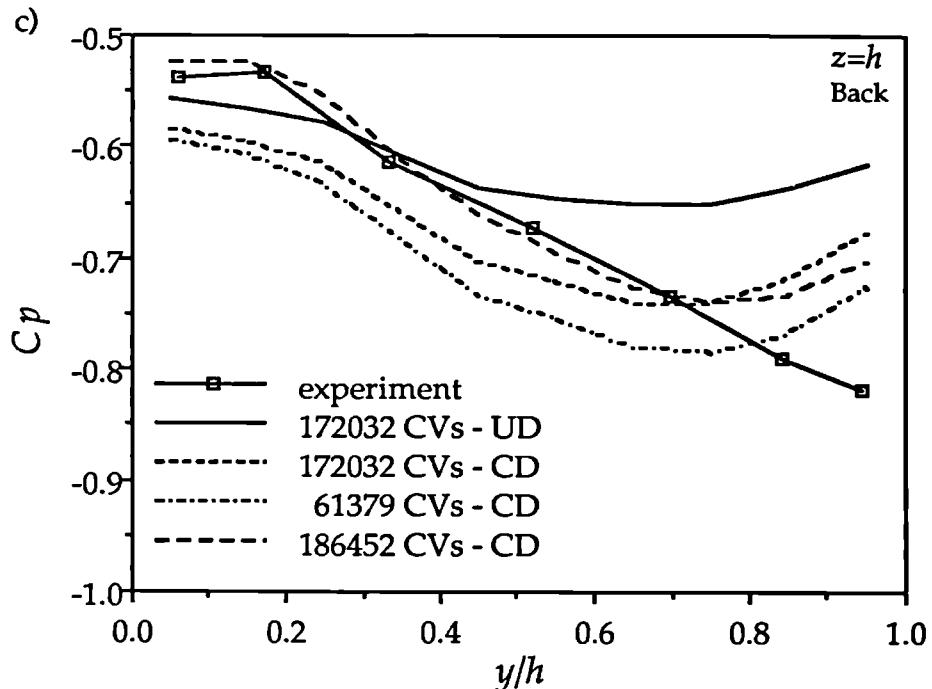


Figure 6.39 (continued)

In general, the agreement between measured and predicted profiles is good at the front surface of the obstacle. In the front plane there is little improvement obtained using the CD scheme or increasing numerical resolution.

The agreement is not so good at the top and side surfaces close to the front edges where the flow separates (Figures 6.39b and 6.41b respectively). The predicted positive pressure gradients in these planes are larger than the measured ones. The trend of these profiles is not influenced by the changes of differencing scheme or the numerical resolution, suggesting that differences between experiment and predictions may be attributed to defects of the $k-\varepsilon$ turbulence model in flows with recirculation (Pope & Whitelaw [1976]) and the treatment of near-wall regions using log-law (Section 3.7.4).

Since it was not possible to produce grid-independent results, it is difficult to judge the quality of the predictions obtained on locally-refined grids as compared to uniformly-refined one. The regions which were selected for refinement by the criterion C2 are in accordance with expectations (Figure 6.37c). The two corners between the walls next to inlet

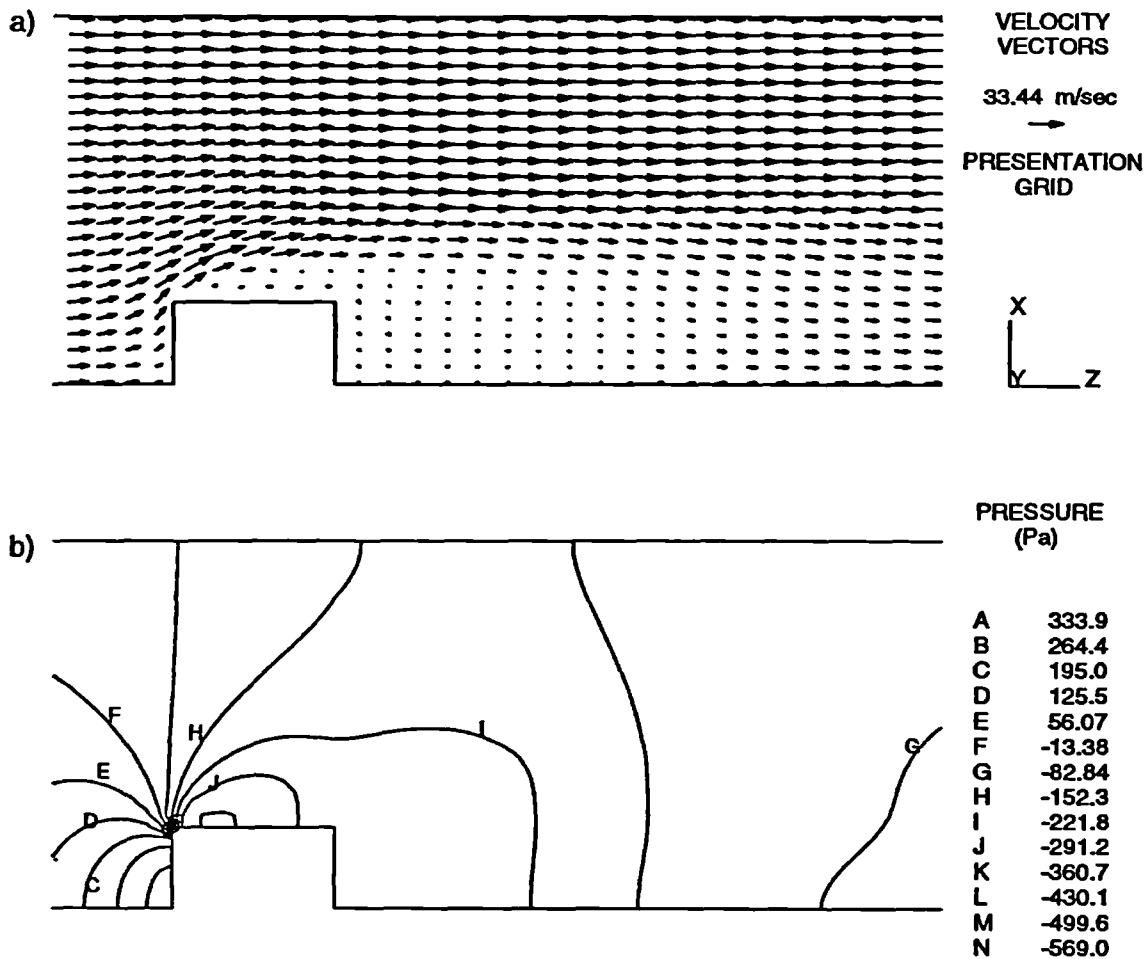
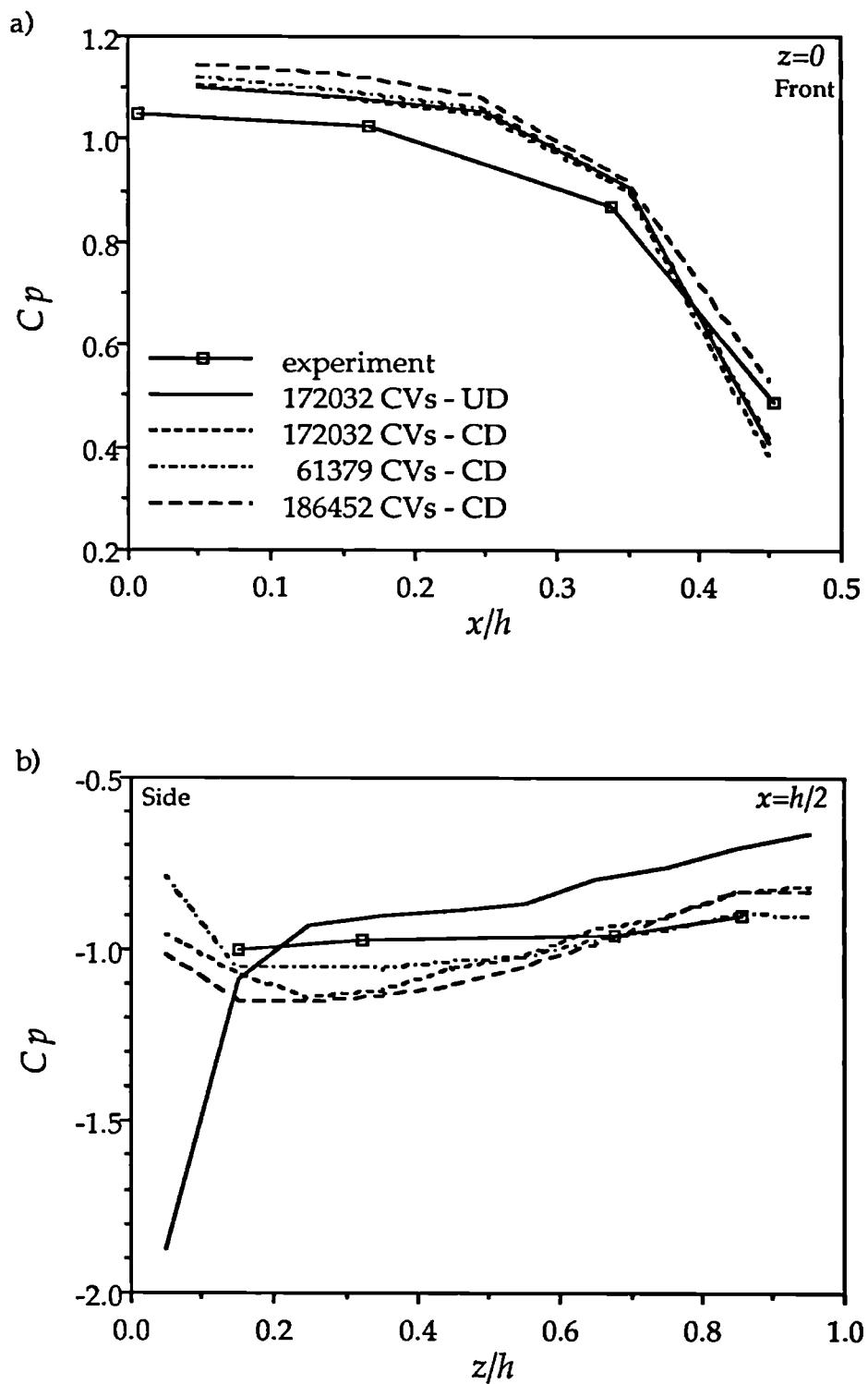


Figure 6.40 Velocity vector and pressure fields in $y=h/2$ plane

(region A in Figure 6.37c) 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 in Figure 6.37c), demanding higher numerical resolution there. Since the flow is disturbed by the obstacle and redirected towards the side and top walls, large variations are present near the corners and between the bottom surface and the symmetry plane immediately downstream of the obstacle (regions C in Figure 6.37c), as the flow tends towards an undisturbed boundary layer.

The maximum and volume-averaged velocity errors on the 61397 CVs mesh were estimated to be 47.26% (near the front and side edges of the obstacle) and 0.65% (volume averaging was performed by multiplying the

Figure 6.41 Pressure variation around obstacle in plane $y=h/2$

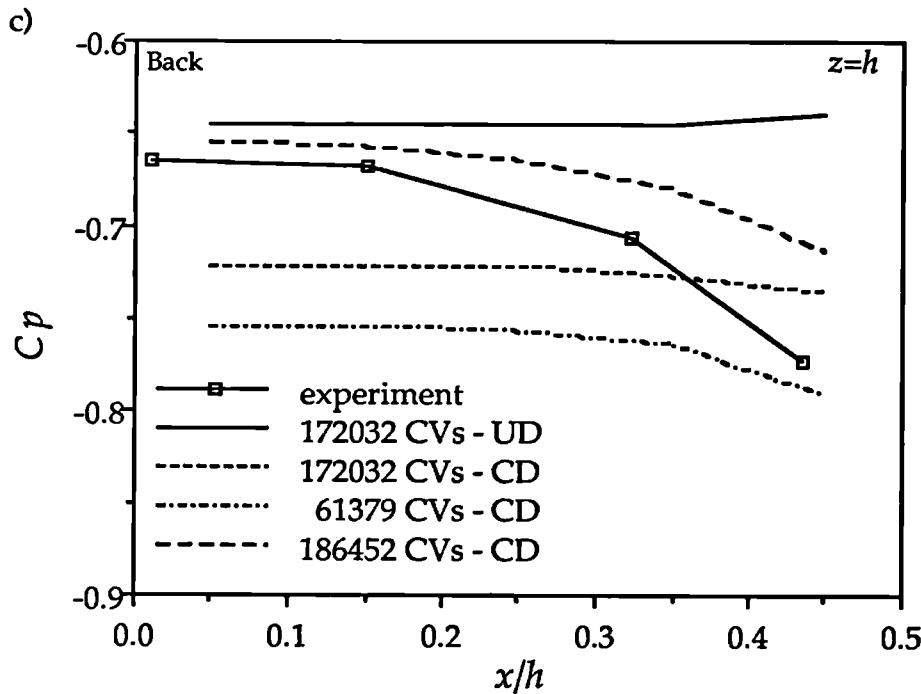
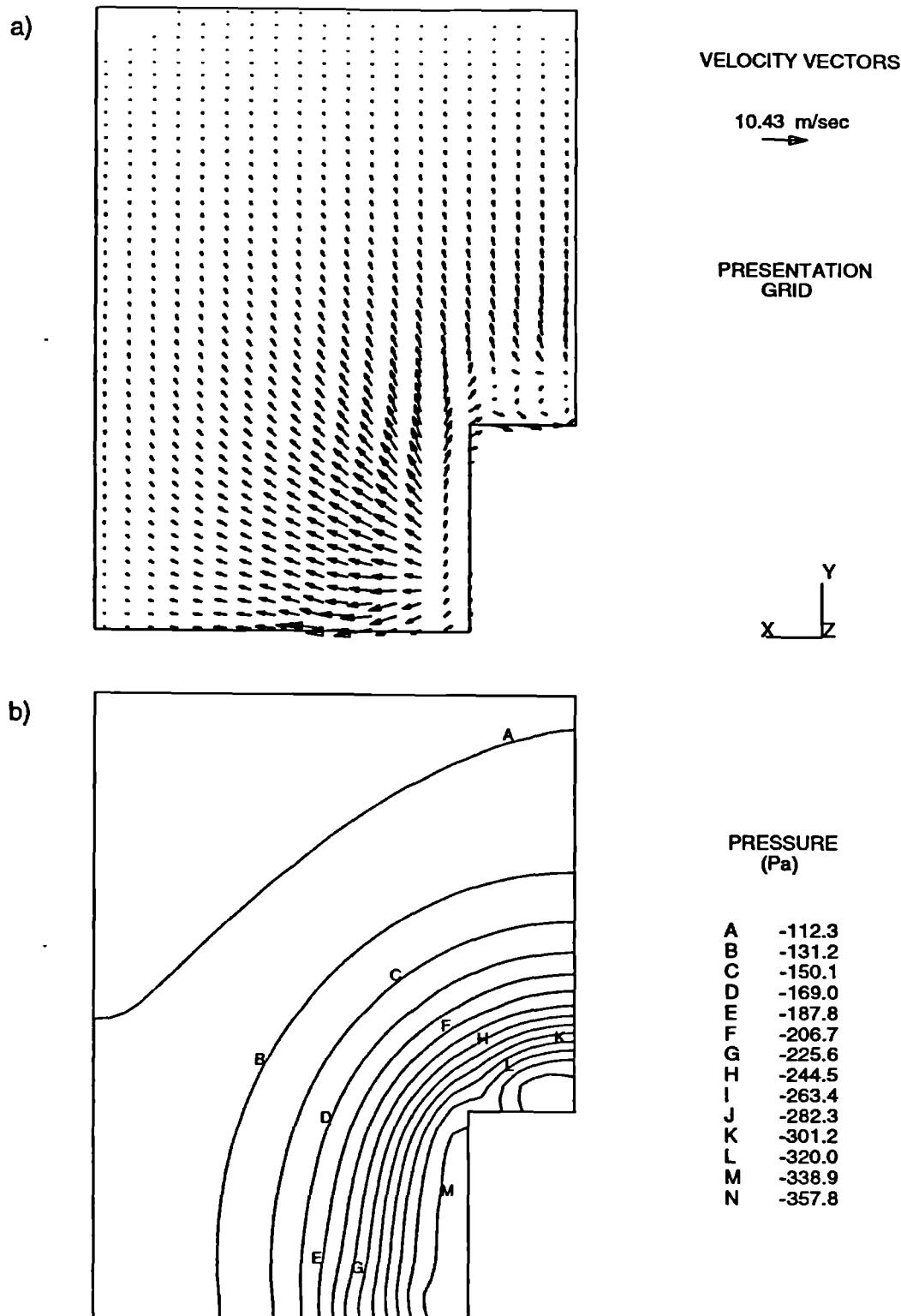


Figure 6.41 (continued)

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% produced a mesh of 186452 CVs reducing the maximum and volume-averaged error to 35.7% and 0.60% respectively. Uniform refinement at the finest level would have needed 1376256. CVs.

The convergence properties of SGM and FMG are given in Table 6.8. The number of fine grid iterations for FMG increases with the number of computational points by an average exponent of 0.06. The same exponent for the SGM method is 0.55. The ratio of CPU times of the SGM and FMG methods is 3.11 with the UD scheme and 4.99 for the CD scheme. FMG combined with local grid refinement has similar convergence properties as FMG applied to uniformly-refined meshes. The number of fine-grid iterations for the locally-refined mesh made of 61397 CVs is almost the same as for the corresponding 172032 CVs uniformly refined mesh.

Figure 6.42 Velocity vector and pressure fields in $z=h/2$ plane

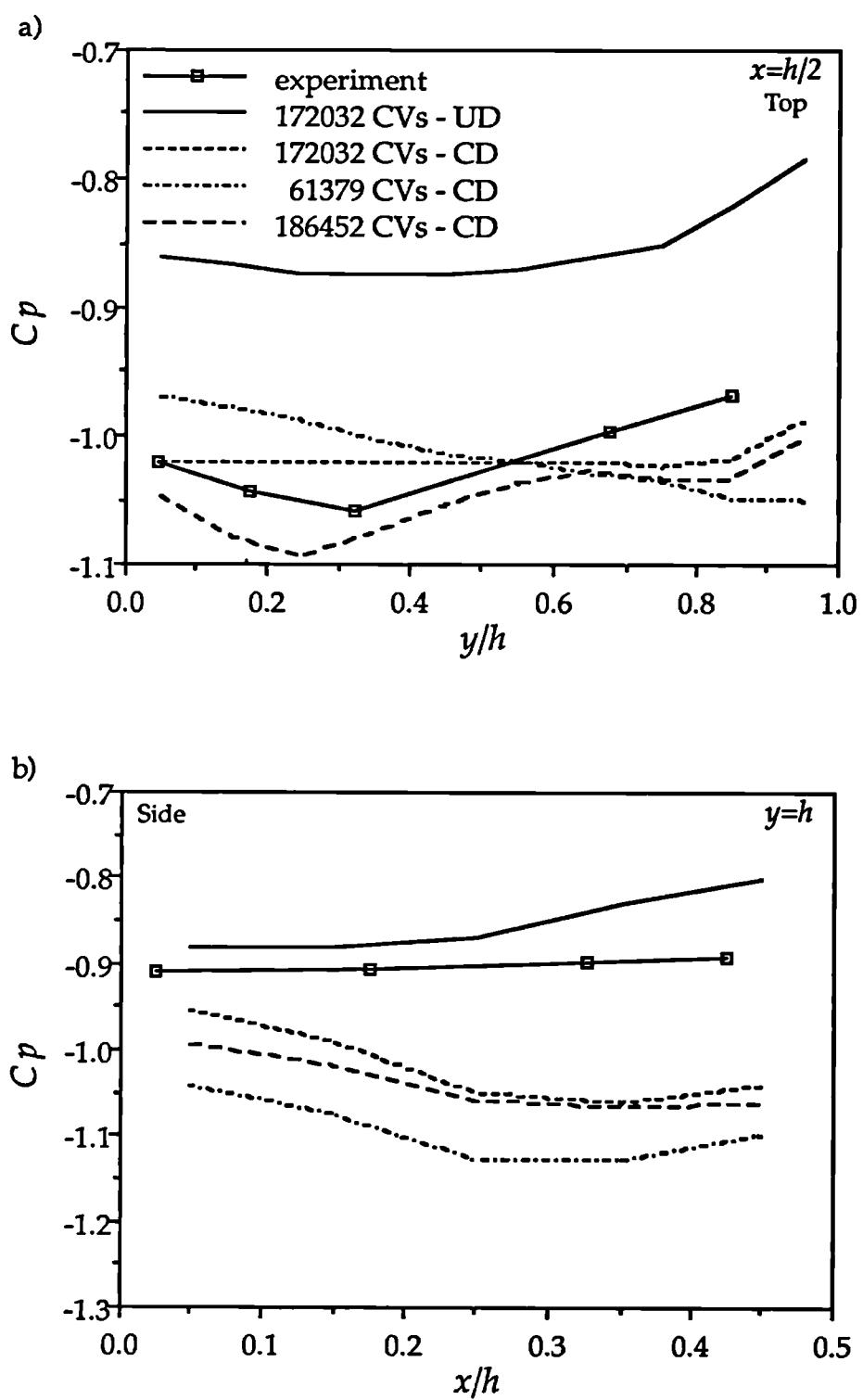


Figure 6.43 Pressure variation around obstacle in plane $y=h/2$

CVs			2688		21504		172032		61397 LGR		186452 LGR	
	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
Central Differencing Scheme												
SGM	74	131.86	164	2678.3	522	79447.	288	19348.	631	107928		
FMG	73	134.82	89	1589.5	101	15898.	82	5326.	99	23805		
SGM/FMG CPU												
		0.98			1.68		4.99		3.63		4.53	
Upwind Differencing Scheme												
SGM	66	114.01	85	1385.1	256	39171.						
FMG	62	120.05	69	1274.3	78	12586.						
SGM/FMG CPU												
		0.95			1.08		3.11					

Table 6.8 Fine grid iterations and computing times required to obtain convergence for flow over prismatic obstacle

The CPU savings due to local grid refinement is 66%. The presented convergence results are similar to those obtained for the two dimensional flow through the orifice plate (Section 6.3). This is encouraging and demonstrates that the present multigrid and local grid refinement techniques can be successfully applied to three-dimensional turbulent flows.

6.5 THREE-DIMENSIONAL TURBULENT SWIRLING FLOW INSIDE A MODEL GAS TURBINE COMBUSTOR

In this section predictions are presented of a three-dimensional swirling flow in a water model of a can-type gas turbine combustor. The combustor model (Figure 6.44) and boundary conditions (Table 6.9) are adopted from the combined experimental and computer study of Palma [1988]. The boundary conditions are given with respect to the polar coordinate system given in Figure 6.44. The combustor model is made up of three different parts: the head, the barrel and the nozzle (see Figure

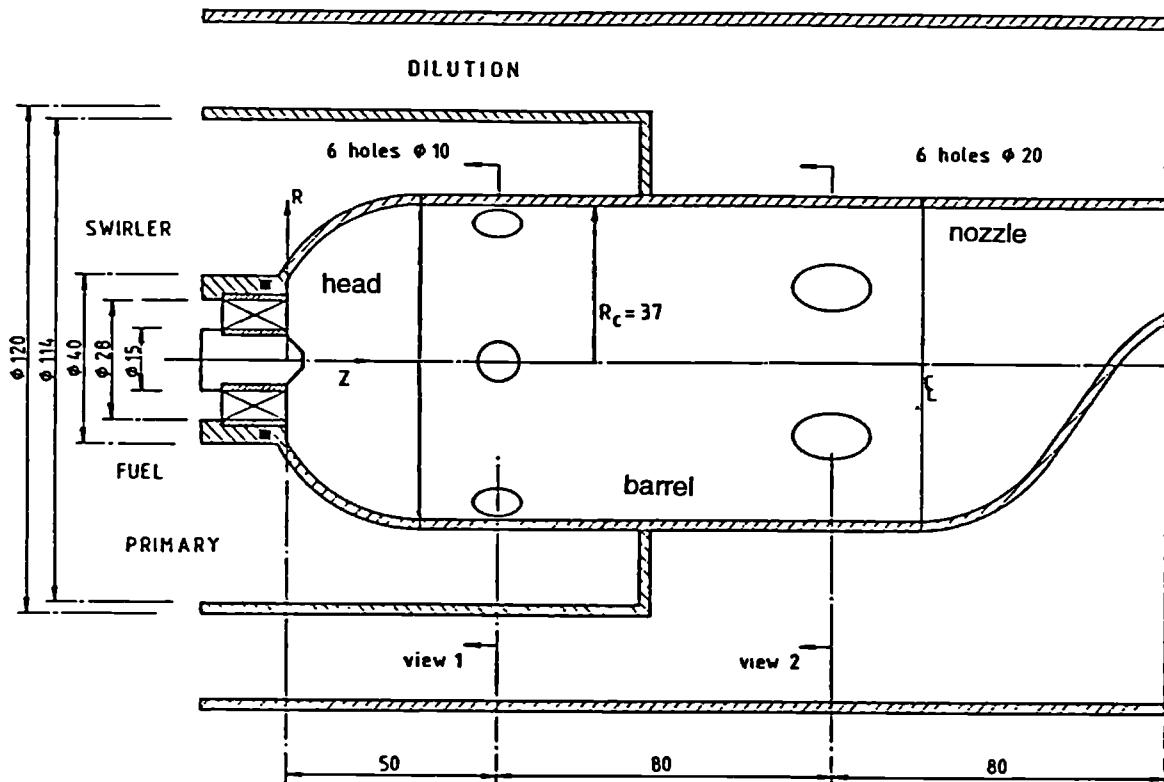


Figure 6.44 Geometry of combustor model

	Axial Velocity (m/s)	Swirl Velocity (m/s)	Radial Velocity (m/s)	Tur. Kin. Energy (m ² /s ²)	Dissipa- tion (m ² /s ³)	Flow Rate (kg/s)
Swirler	1.25	0.95	0.00	1.50×10^{-2}	3.06×10^1	0.543
Primary	0.00	0.08	-2.22	1.24×10^{-1}	4.37	1.040
Dilution	0.20	0.00	-0.85	1.12×10^{-2}	5.96×10^{-2}	1.600

Table 6.9 Inlet boundary conditions for the combustor model at swirler, primary and dilution holes.

6.44). The barrel contains six 10 mm primary and six 20 mm dilution holes, 80 mm apart and staggered by half a pitch with respect to each other (Figure 6.45), but otherwise equally spaced around the cylindrical barrel. The geometry and the flow (Figure 6.47) are complex enough that the exercise will fully demonstrate the flexibility and efficiency of numerical techniques developed in this work.

The coarsest and two levels of uniformly-refined mesh used for the calculations are presented in Figure 6.45. Refinement was not performed in the nozzle section of the combustor since this area is far downstream from the regions of interest. The initial grid has 342 CVs, and its geometrical boundaries are described by surface splines, such that when refinement is performed the outermost vertices of the mesh always follow the geometrical surface description.

This test case well demonstrates the strategies and techniques which were summarised in Section 5.6. The generation of a coarse mesh is much less labour-intensive than a fine one. The cells comprising the coarsest (initial) mesh are effectively 'blocks' or templates which characterise the geometry at the crudest level, and whose refinement is simple and is driven by an error estimate. Their number will depend on the complexity of the geometry. In this way, an automatic, adaptive and accurate mesh generation is possible, offering a powerful and flexible engineering design tool.

The final locally-refined mesh obtained automatically starting from the coarsest one, is presented in Figure 6.46. A blend of the upwind and central differencing schemes (Section 3.4) was used, with the blending factor γ set to 0.9. The refinement was controlled by the C2 criterion, and the normalisation was based on the values of the velocity magnitude, k and ε at the primary holes (Table 6.9). After a doubly-uniform refinement the estimated error was still larger than the prescribed error tolerance T^* of 1% (as described in Sections 5.5 and 5.6) in the whole domain. However, because of computer memory limitations further refinement was not possible, and for the purpose of demonstrating the methodology, the error tolerance T^* is set to 20%. The resulting mesh is presented in Figure 6.46.

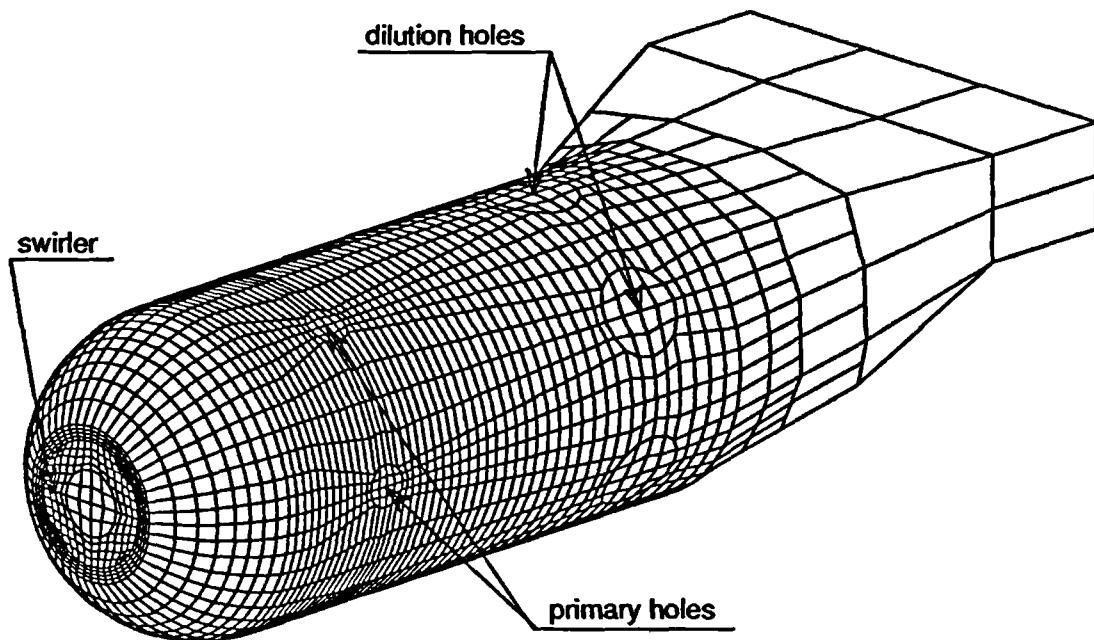
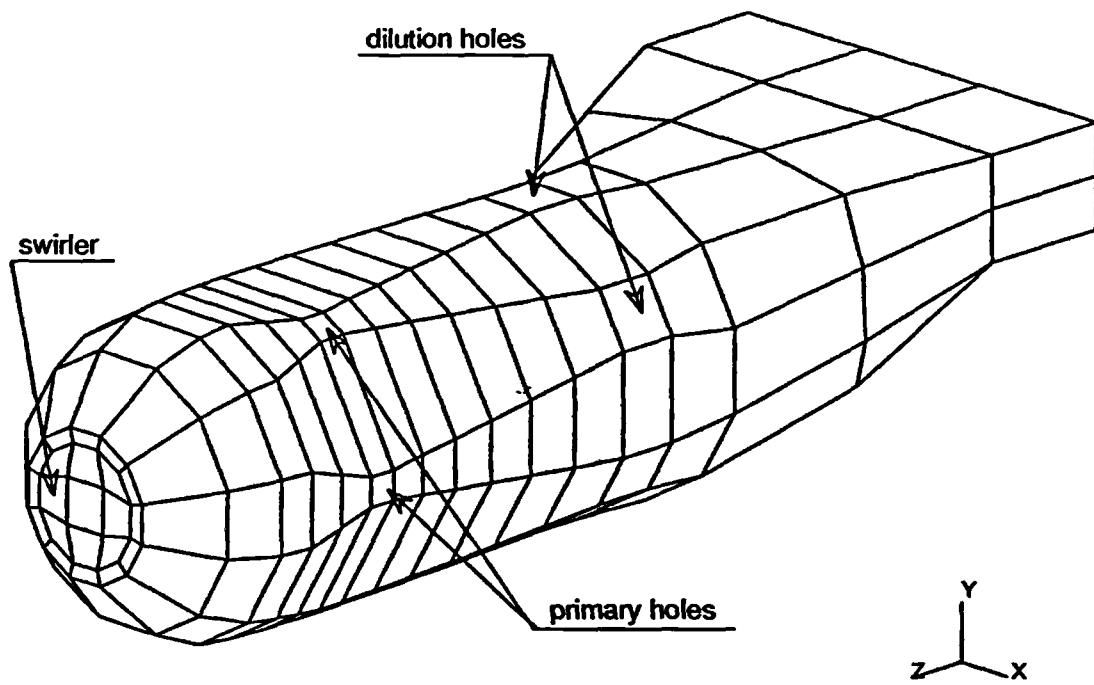


Figure 6.45 The initial and a doubly uniformly refined (apart from nozzle) numerical meshes

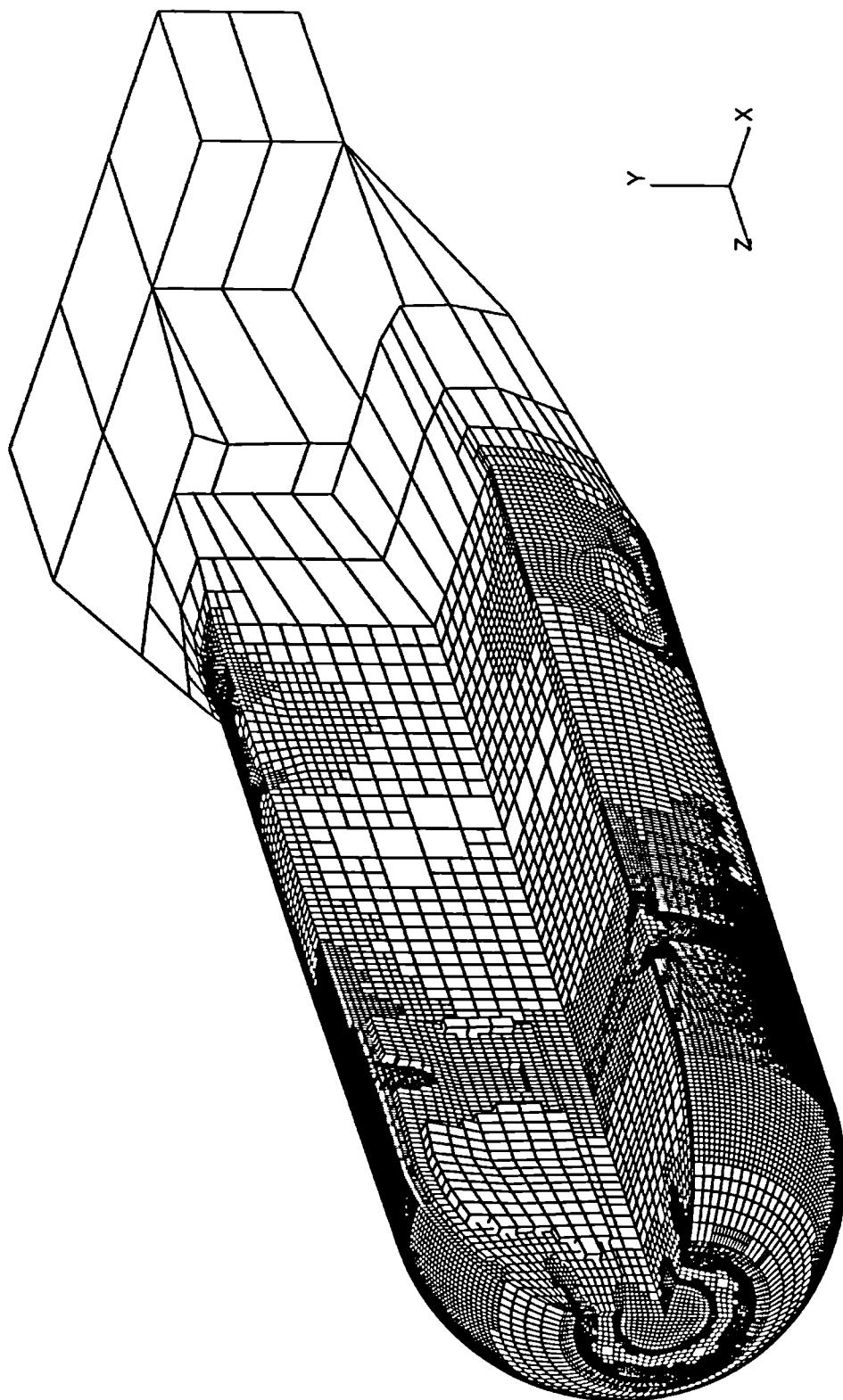


Figure 6.46 Locally refined mesh - 248316 CVs

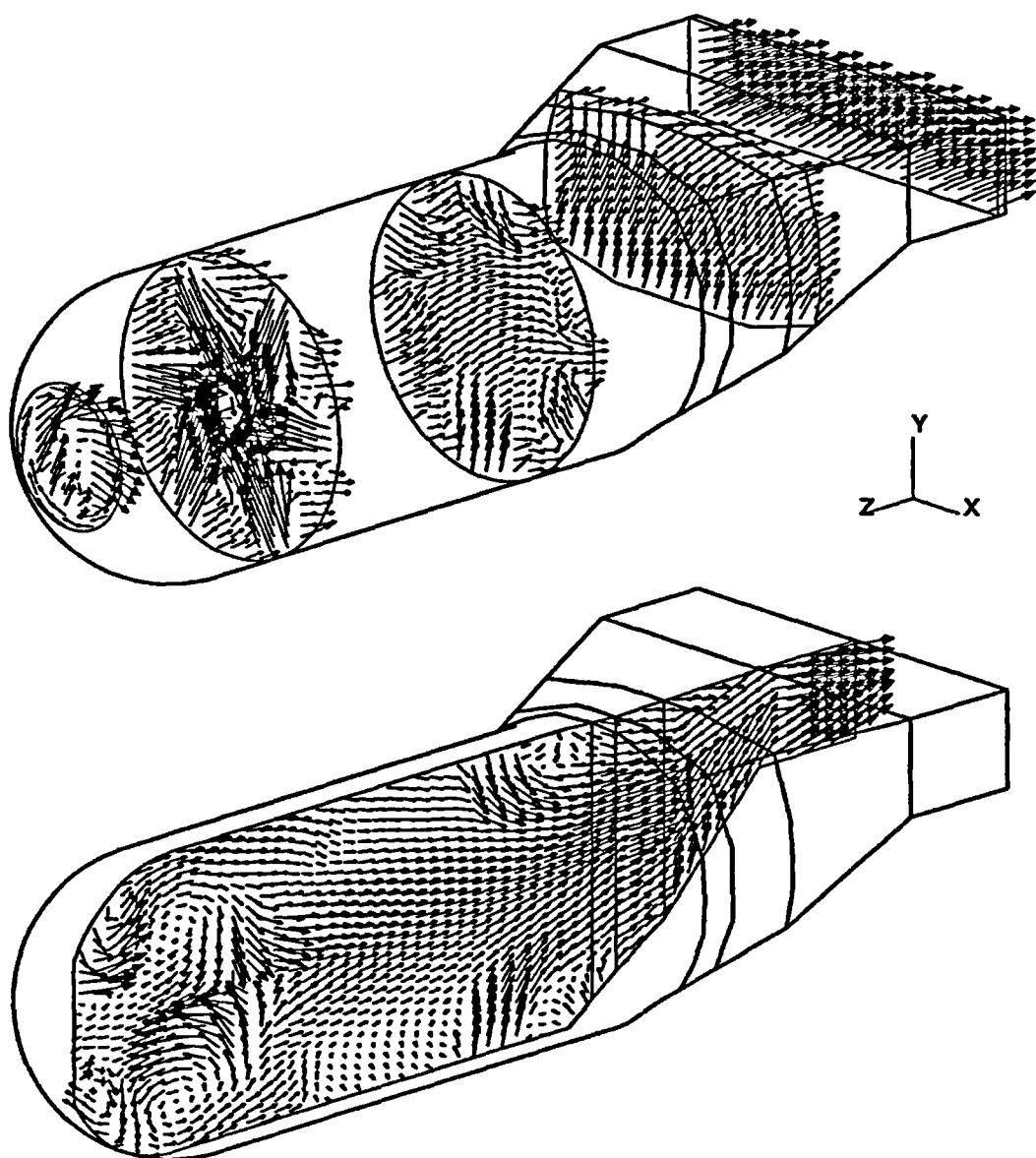


Figure 6.47 Overall flow pattern of model-combustor flow

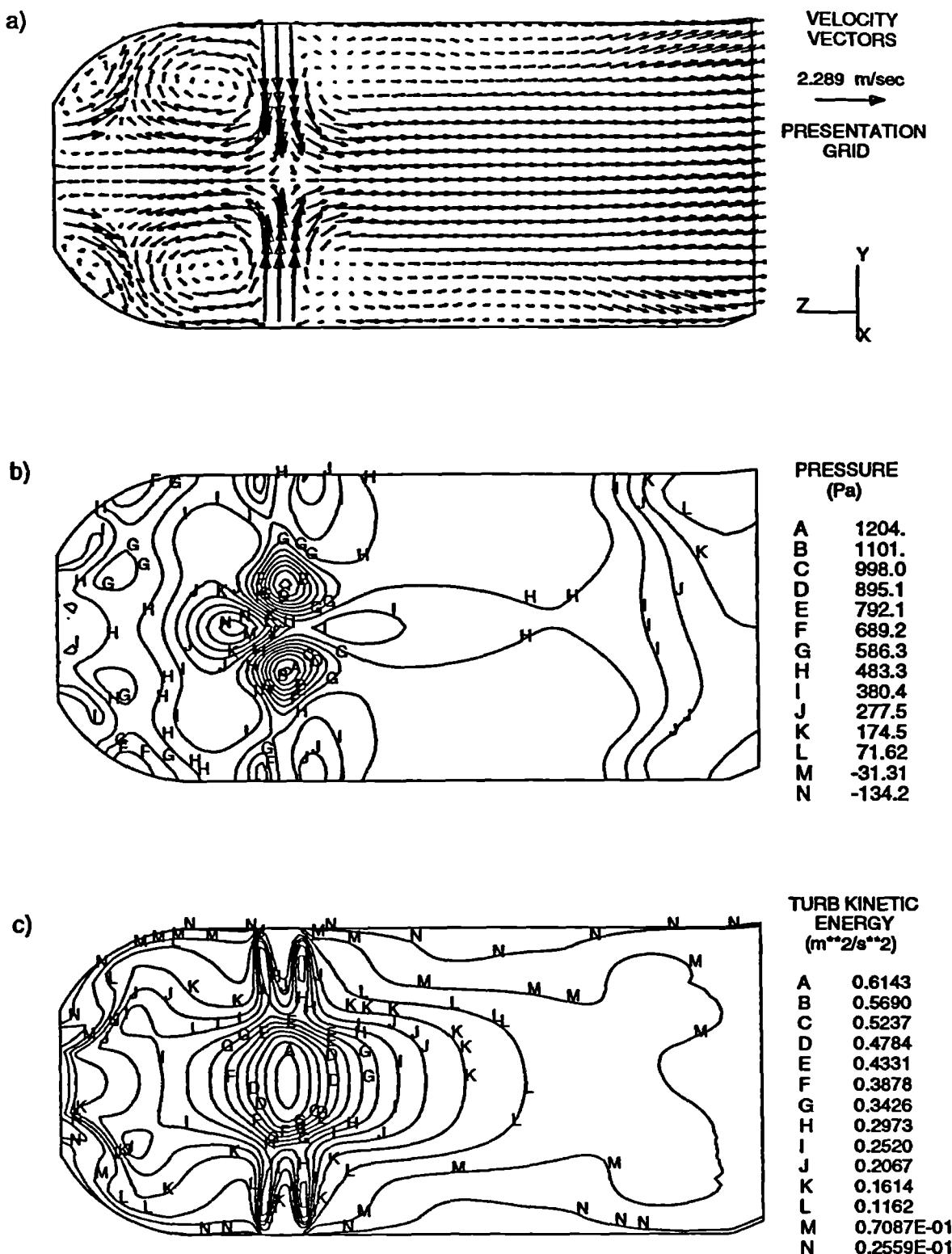


Figure 6.48 Velocity a), pressure b) and turbulence kinetic energy c) fields in the longitudinal plane containing the axes of the primary holes

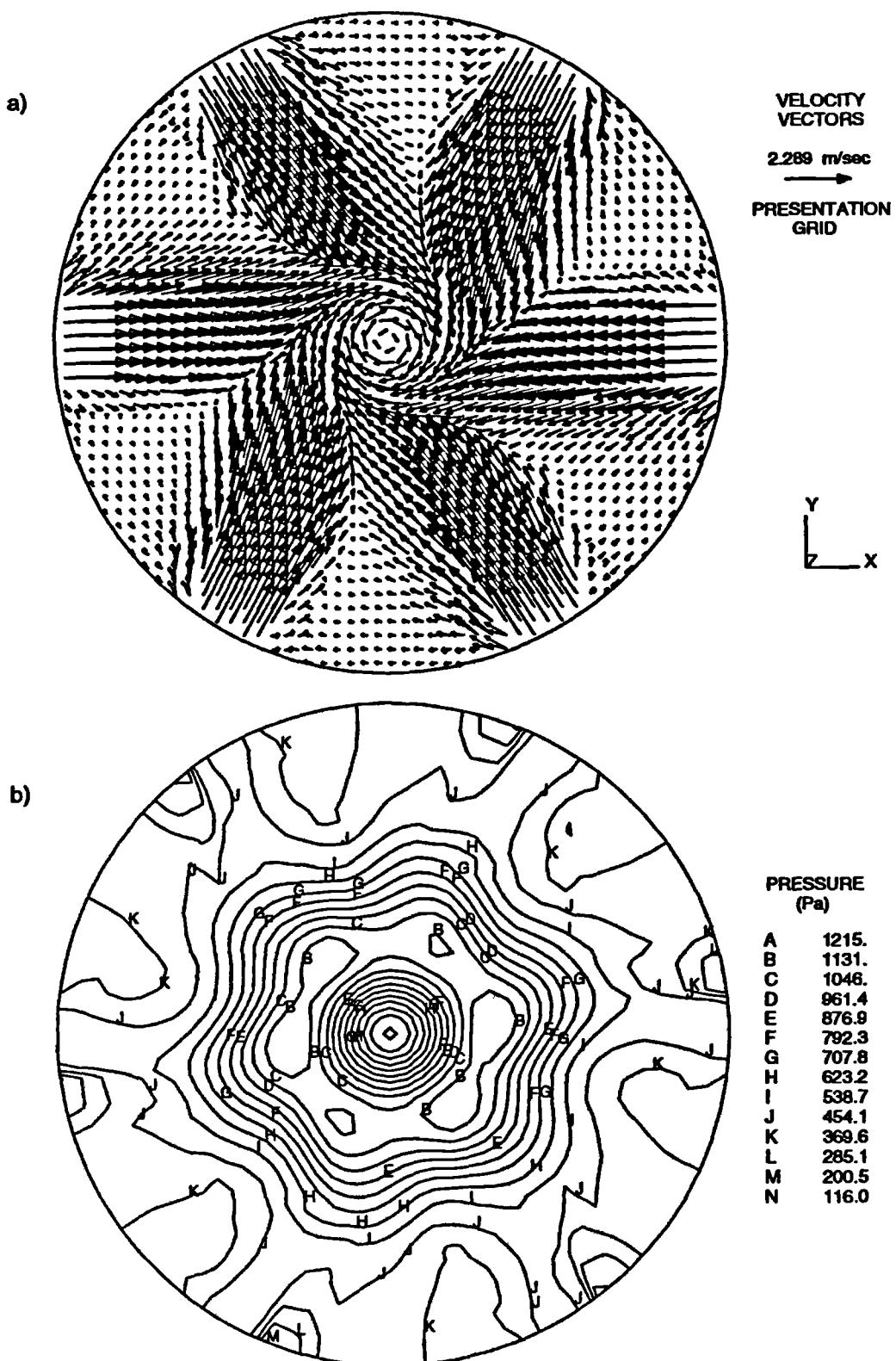


Figure 6.49 Velocity a) and pressure b) fields in the cross-section plane containing the axes of the primary holes

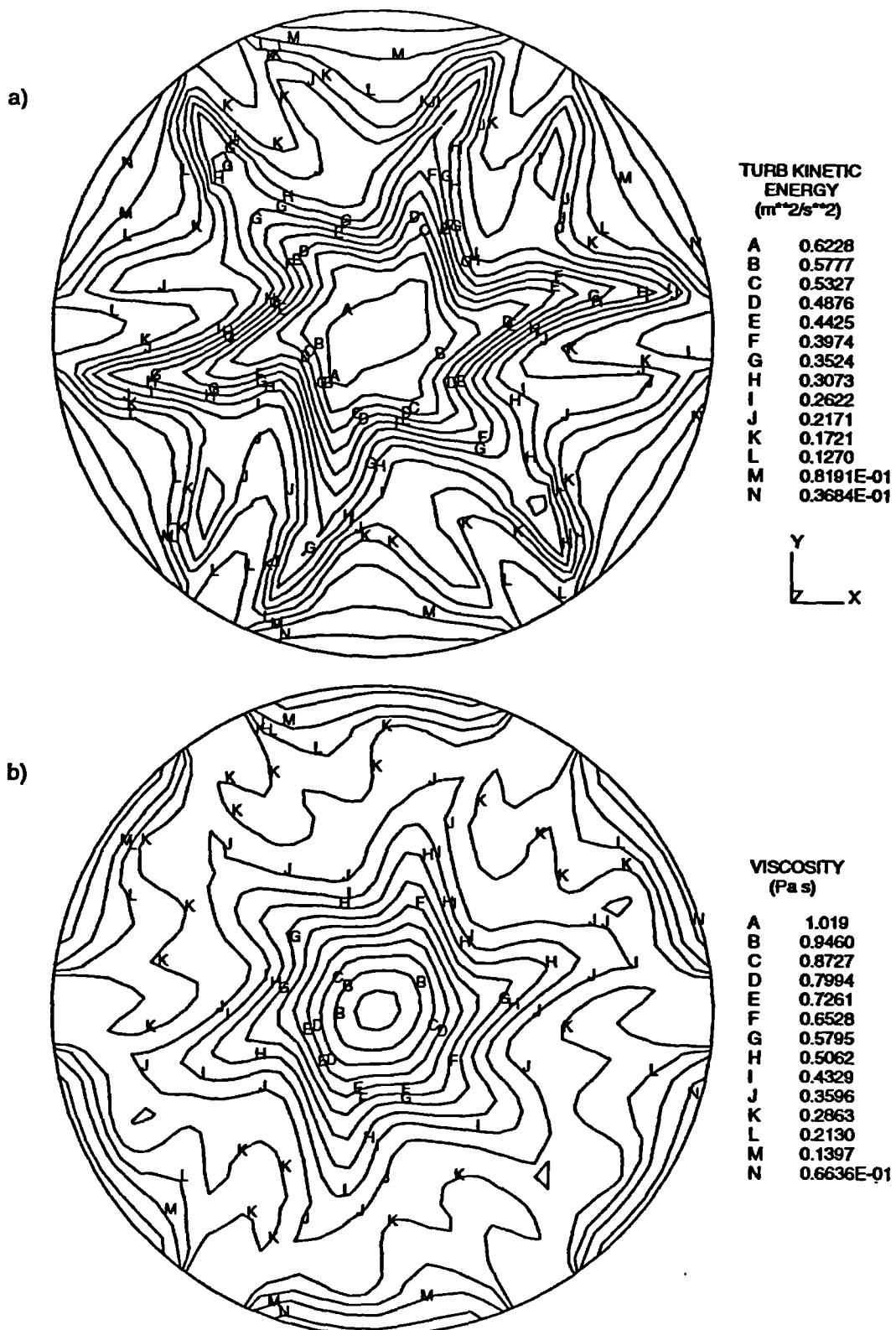


Figure 6.50 Turbulent kinetic energy a) and effective viscosity b) fields in the cross-section plane containing the axes of the primary holes.

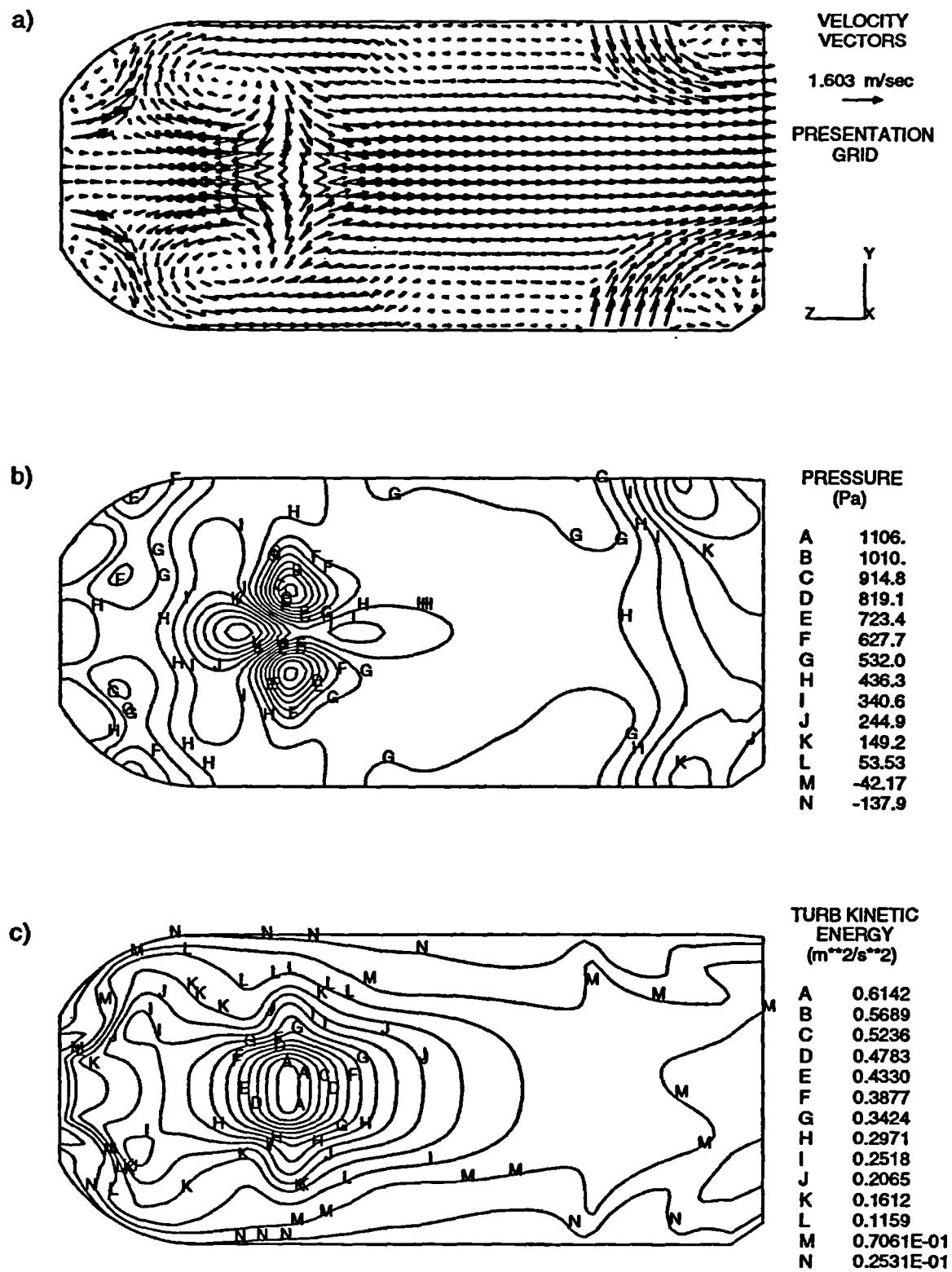


Figure 6.51 Velocity a), pressure b) and turbulence kinetic energy c) fields in the longitudinal plane containing the axes of the dilution holes

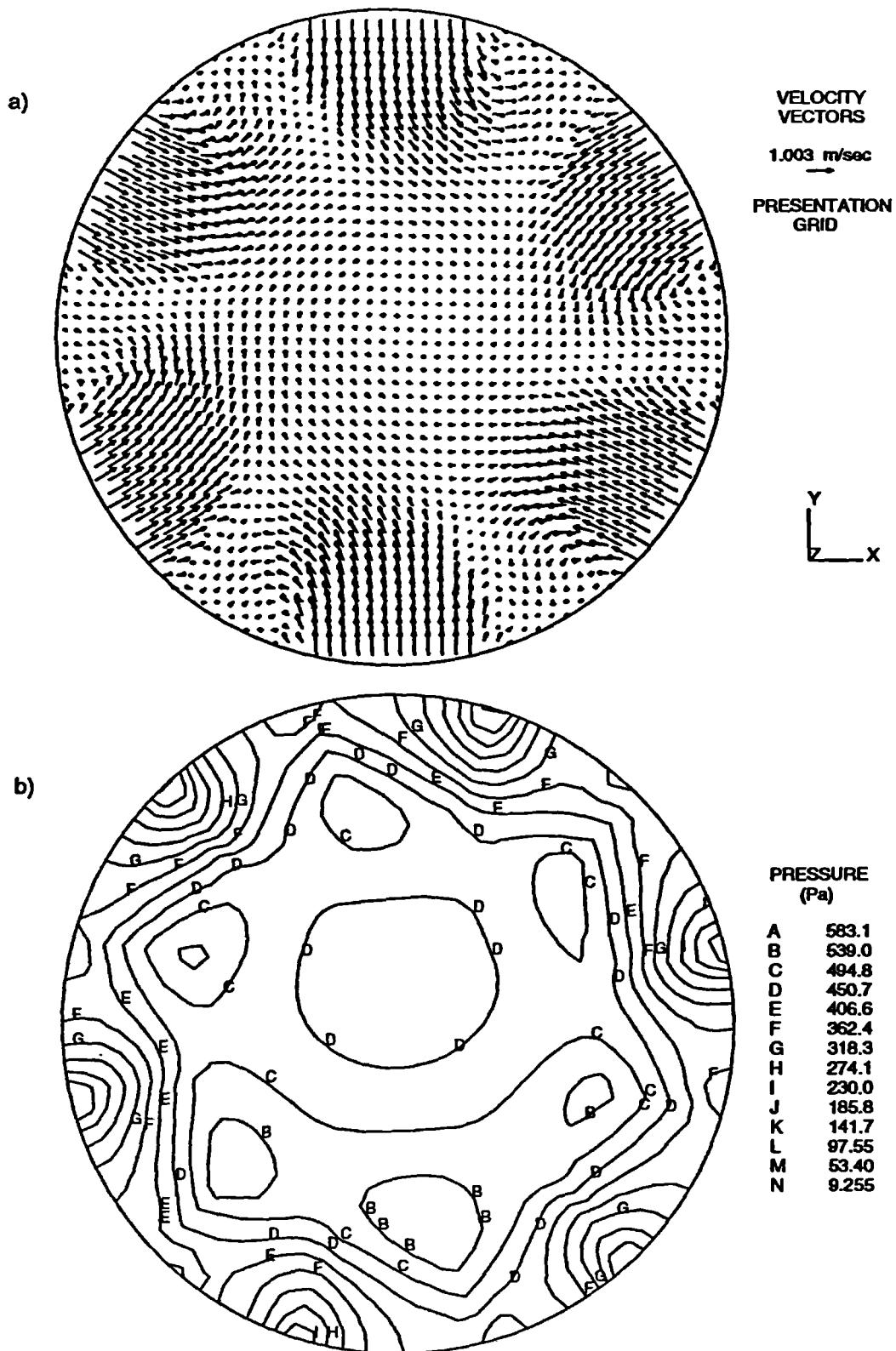


Figure 6.52 Velocity a) and pressure b) fields in the cross-section plane containing the axes of the dilution holes

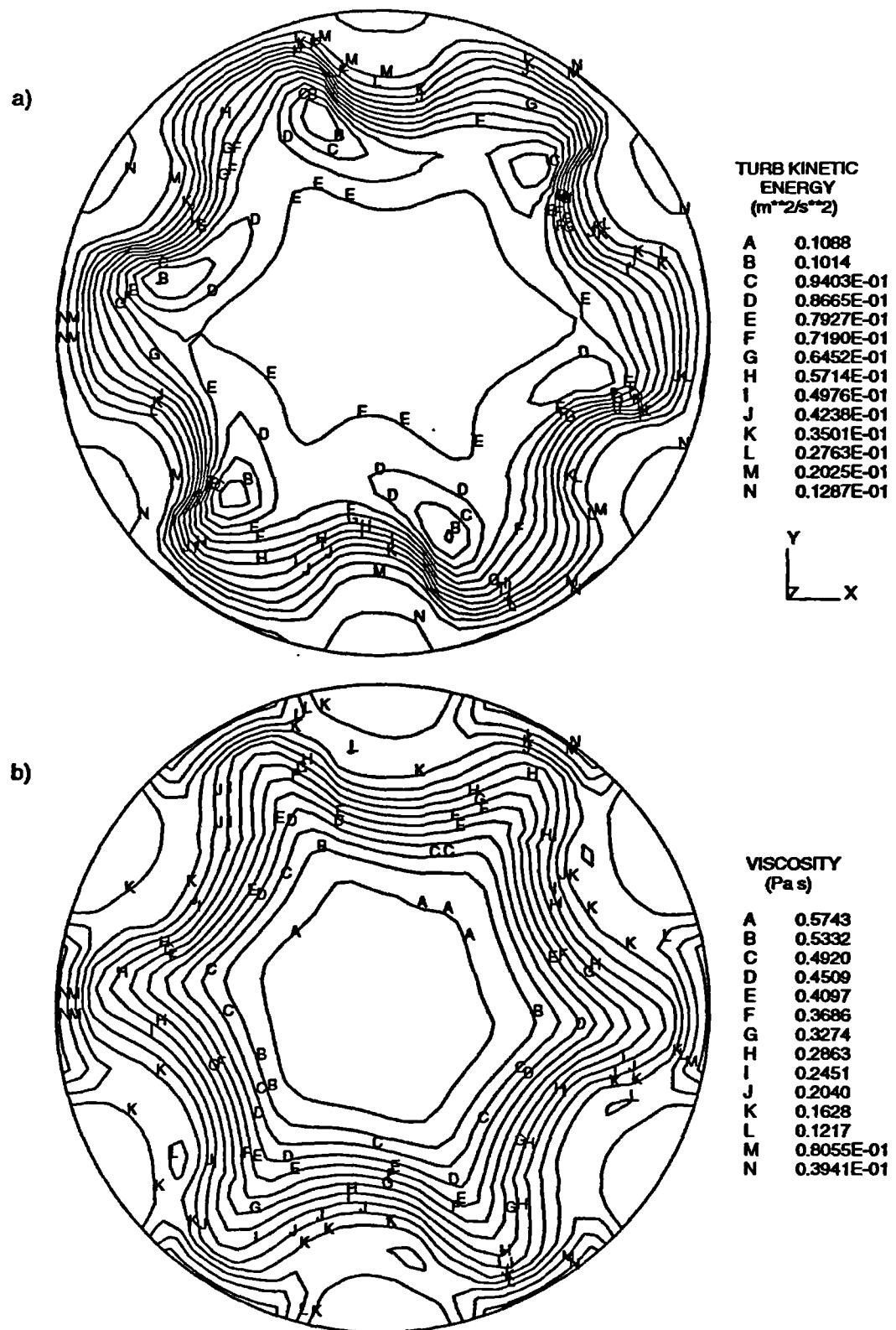


Figure 6.53 Turbulent kinetic energy a) and effective viscosity b) fields in the cross-section plane containing the axes of the dilution holes.

A selection of predicted results obtained on a locally refined mesh and then mapped onto a uniform Cartesian presentation grid are given in Figures 6.47 to 6.53. Firstly, Figure 6.47 gives a qualitative overview of the predicted 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 towards the centre line. The velocity, pressure and turbulent kinetic energy fields in a longitudinal plane which contains the axes of primary holes are presented in Figure 6.48. The distributions of the same variables in the cross-sectional plane containing the axes of the primary holes are given in Figures 6.49 and 6.50. All these fields are cyclically repeating in the zone of the swirler and the primary holes. The fields of the dependant variables in longitudinal and cross-section planes containing the axis of dilution holes are given in Figures 6.51 to 6.53.

A parameter observed to respond sensitively to both turbulence modelling and numerical discretisation is the axial centreline velocity, as noted by Lin and Leschziner [1989] in their computational study. They have compared their results with experimental data of Koutmos [1985]. Figure 6.54 depicts four variations obtained with different grid densities, compared with the experimental and numerical data of Palma [1988]. The uniform grids were third and fourth multigrid levels, 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. The next level of local grid refinement produced a grid composed of 248316 CVs, which is presented in Figure 6.46. The velocity profile obtained on this grid exhibits more strongly an anomalous trough upstream of the primary holes. Lin and Leschziner [1989] have linked this to the large axial pressure gradients accompanying a correspondingly-large axial variation in the swirl velocity in the vicinity of the centreline. They performed calculations using the standard $k-\varepsilon$ eddy-viscosity model and two variants of the Reynolds-stress transport model (RSTM) of turbulence. The $k-\varepsilon$ 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 $k-\varepsilon$ model has similar poor behaviour in this region as the RSTM.

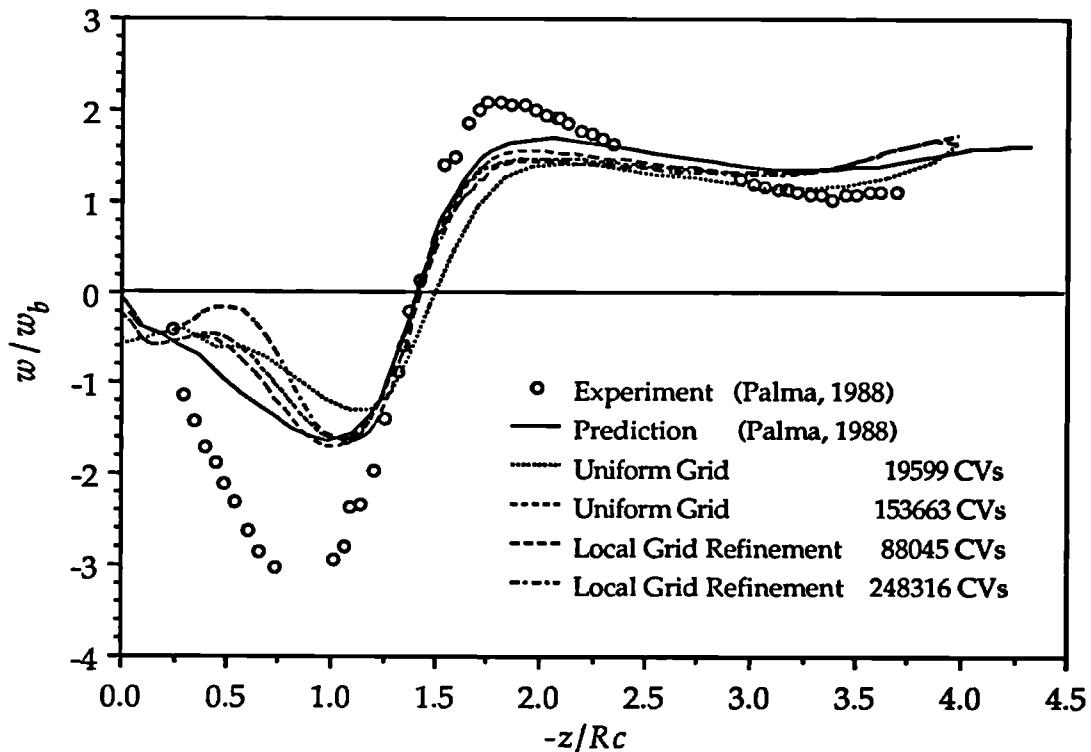


Figure 6.54 Axial velocity along the combustor centreline

The predictions of Palma [1988] (Figure 6.54) were obtained using a uniform grid composed of 39600 CVs to mesh one-sixth of the combustor, assuming cyclic repeatability. His mesh is finer (6×39600 CVs would be necessary to mesh the whole combustor) than the finest uniformly-refined one used in the present study. The use of local grid refinement in the present study, allowed the concentration of computational points in regions where the large source of error was detected. Numerical resolution in those regions is much higher than Palma had used for his predictions. However, since the refinement tolerance T^* was so high (a tolerance of 1% would have led to meshes of well over 2×10^6 cells), the final solution is probably still quite far away from the grid-independent one in some regions.

Error-led local grid refinement (Figure 6.46) acts again in accordance with expectations. More computational points are concentrated in regions around swirler and primary and dilution jets where the strong shearing flows exist. The relatively uniform cores of the jets are not refined so much, while refinement takes place in the mixing layers of the jets where strong velocity gradients are present. The refinement takes place in near-wall regions as well.

Table 6.10 presents the fine-grid iterations and computing times for this case to converge. The speed-up achieved is up to 3.26 for multigrid performed on four uniformly-refined grids. The use of the local grid refinement does not have a big influence on the multigrid convergence rate. However, the performance relative to the SGM method gets worse, since the number of fine grid iterations necessary to achieve convergence using SGM is reduced compared to the uniformly-refined mesh. There is an increase in the number of fine-grid iterations for FMG with mesh refinement, as was noted for other turbulent cases, and the possible reasons will be discussed in the next section.

CVs		2547		19599		153663		88045 LGR		248316 LGR	
		Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU	Iter	CPU
SGM	112	173.51		236	3137.8	649	74553	468	36406	562	137788
FMG	101	182.30		143	2296.0	168	22835	152	13991	128	40325
SGM/FMG CPU											
		0.95		1.37		3.26		2.60		3.42	

Table 6.10 Fine grid iteration and computing time required to obtain convergence for flow inside combustor

It is rather difficult to comment on the savings in computer memory achieved by local grid refinement. Because of the memory limitations of the available computer hardware, it was not possible to obtain the solutions of desired numerical accuracy, either with uniformly- or locally-refined meshes. Since the flow exhibits very complex features, it is difficult to conclude from the similarity of the profiles obtained using 153663 CVs uniformly and 88045 CVs locally refined mesh, that the same accuracy has

been achieved, with a reduction of the number of control volumes of 43%. However, the results are encouraging, indicating clearly the benefits which come from combining local grid refinement and multigrid techniques.

6.6 CLOSURE

In this chapter, the efficiency of the present solution procedure has been assessed for the following problems: the two-dimensional laminar driven cavity flow at Reynolds numbers of 100 and 1000; two-dimensional turbulent flow through an orifice plate; three-dimensional turbulent flow over a surface-mounted cubical obstacle; and three-dimensional turbulent flow inside a gas turbine combustor. All the cases contain local regions of steep gradients and oblique flow which could give rise to severe numerical errors: thus they provide a good test for local grid refinement and overall algorithm efficiency.

The multigrid convergence rates for laminar two-dimensional driven-cavity flow was found to be in agreement with multigrid theory (i.e. convergence rate independent of mesh density), and similar to other reported results (Vanka [1986], Gaskell et. al [1987], Barcus et. al [1987], Smith [1990]). Turbulence tends to reduce the effectiveness of the multigrid scheme. A similar conclusion was arrived at from the numerical experiments performed by Lien [1992]. However, although the number of fine grid iterations increases slightly with the increase of grid number, the savings in computer time obtained (3 to 9 times reduction) are nonetheless encouraging, considering the complexity of the physical situations considered and the relative coarseness of the meshes used to resolve these flows.

Simulation of turbulent flows using wall functions is a problem which can be an important aspect for multigrid efficiency. The non-linear nature of the wall-functions means that rather fine numerical resolution is needed before their spatial variation will appear to be smooth. This means that when the wall functions are used as in the present study, the coarsest

mesh used for multigrid cannot be too coarse, otherwise the boundary conditions cannot respond 'accurately' to the changes of dependent variables in the near-wall region. In this case the coarse grid corrections may not efficiently drive the fine grid solution towards convergence. In addition, the use of wall functions limits the mesh resolution in near-wall regions, thereby not allowing refinement everywhere in the computational domain. Thus, strictly speaking, grid independent solutions are not accessible.

Turbulent kinetic energy and its dissipation rate are two quantities which are dependent very much on the mesh resolution. Since the field of eddy viscosity depends on these two fields, the turbulent viscosity field of the flow may look quite different on the coarsest and the finest meshes at an early stage of the multigrid cycle. However, this early stage can be critical in determining the multigrid efficiency or even hindering convergence altogether. At present, this limits the minimum mesh resolution which can be used. Otherwise, the representation of the fine-grid solution on a very coarse mesh may be so poor and that the corrections obtained on that mesh will be of very little use. The way of dealing with this problem in the present study has been to obtain the field of eddy viscosity on the coarser meshes by interpolating the values of eddy viscosity from the finest one. Although this appears to be a good engineering approach, it does not completely solve the problem. The eddy viscosity field develops as velocity field develops, so they simultaneously affect each other. The practice of fixing the eddy viscosity on a coarser grid therefore introduces some form of under-relaxation without taking the full advantage of the coarse grid iteration.

'Standard' multigrid solvers are known deal effectively with algebraic systems describing elliptic problems. For the problems which do not have elliptic nature or whose partial differential operator has a nonelliptic component, the performance of standard multigrid methodologies, like the one described in the present study, tends not to be optimal. There are recently published works which address this problem (Brandt & Yavneh [1992,1993]), The proposed techniques for more closely approaching the optimum multigrid performance for this class of problems, are specifically designed to solve a limited class of flows (e.g. entering or recirculating

flows). These studies have enriched the knowledge about the cause of the poor multigrid behaviour and show some ways of how to deal with them and get optimum performance, but there is much more work to be done before their lessons can be applied to the kinds of application considered in the present work. However, those who do not seek optimum performance, can still find great potential in the available multigrid algorithm, that offer considerable savings in CPU time, as can be seen from the results of this chapter.

For the driven-cavity case the accuracy of locally refined solutions was assessed by comparison with full fine-grid solutions shown to be grid independent. Comparable accuracy was achieved with only a fraction of the total number of cells.

As can be seen from the plotted fields of the solution error, the novel error estimation technique has good qualitative behaviour in comparison with the exact error distribution. Maximum and averaged error estimates were compared for the cavity flow at $Re=100$, to give a measure of the quantitative value of these estimates. The novel error estimation technique, as well as the one based on Richardson extrapolation, underpredict the average solution error, while the effectivity index approaches one as the mesh is refined. Solving the transport equation of the solution error, whose sources are the estimated tau errors, tends to increase the estimated average and maximum error.

The novel error estimation technique presented in Sections 5.4.2 and 5.4.3 was found to be a useful refinement indicator. Error estimation is a challenging task even in the case of the transport of a single scalar by convection and diffusion. Since fluid flow is governed by a set of coupled non-linear equations, the solution error of each of the dependent variable field depends not only on the numerical resolution of that particular field, but also on the errors present in the fields of all other coupled variables which are involved in the calculation. This makes the quantitative error estimation in fluid flow calculations even more difficult. However, local refinement based on an estimate of truncation error behaved in accordance with intuitive expectations, concentrating computational

points in regions where steep variations of the dependent variables occurred.

The influence of different iterative solvers for the solution of the systems of linear algebraic equations generated within the SIMPLE algorithm has been investigated. The main conclusion is that the overall convergence rate of a multigrid method which is based on the SIMPLE algorithm as a smoother is limited by the convergence properties of the base algorithm for interequation coupling rather than on the solver used for the solution of the linearised equation sets. Gauss-Seidel iteration was selected for solution of all the non-symmetric equations. The incomplete Cholesky preconditioned CG solver was chosen for the solution of the continuity (pressure-correction) equation, since it does not exhibit the desired diagonal dominance, and classical iterative techniques therefore sometimes struggle to achieve convergence.

The first-order prolongation operator has proved to be better in MG calculations than the zero-order one in the case of the two-dimensional driven cavity flow at $Re=1000$. The differences between the two practices were small for $Re=100$, due to the smooth variation of the dependent variables. Since the additional computational cost for the implementation of the first-order prolongation operator is not large, this practice was adopted for all calculations.

This completes the assessment of the present solution procedure. In the next chapter, an overall summary of the thesis findings will be given, together with some recommendations for future research.

Chapter 7

Summary and Conclusions

The aim of this study was to improve the efficiency of computational fluid dynamics calculations in complex geometries of the kind encountered in everyday engineering problems. The approach is to enable solutions free from the numerical error to be achieved economically by employing local mesh refinement to resolve regions of steep gradients, and using a multigrid method to accelerate the convergence of the basic iterative solution technique. In order to identify regions requiring local refinement, and to assure the quality of the numerical solution, it was necessary to devise and investigate the use of error indicators .

Since mesh generation, which entails the accurate surface description of complex computational domains and the efficient local grid refinement, is a much easier task if one uses unstructured meshes made of cells of 'arbitrary' topology, another objective was to devise a discretisation practice which is independent of cell topology.

7.1 DISCRETISATION

Because of the reasons mentioned above, discretisation of the computational domain is based on a use of unstructured meshes made of cells of 'arbitrary' topology.

The integral form of the governing equations of Newtonian fluid flow accompanied by the k - ε model of turbulence are applied on a finite number of control volumes (cells) subdividing the computational domain, and then discretised using a practice which has been developed in the present study. The novel discretisation is capable of dealing with non-orthogonal meshes made of cells of arbitrary topology. It is performed in real space, using global Cartesian coordinates in a manner which is easy to understand and implement. The least squares technique, which is used to determine the coefficients of an assumed spatial distribution of the dependent variable, has proved to be an elegant way of dealing with cells of arbitrary topology, and at the same time to be capable of delivering the desired accuracy even on non-uniform and non-orthogonal grids. In the present study a linear polynomial fit was chosen, so that its coefficients could be determined by visiting only the nearest neighbours of a cell which has the most primitive topology (tetrahedral in 3D or triangular in 2D). One of the formulations presented in this study gives second-order accuracy in modelling diffusive transport even on a non-uniform mesh. The present practice allows a consistent and uniform discretisation treatment throughout the entire computational domain including the interface between refined regions and the rest of the computational mesh.

In order to avoid large numerical dispersion, convection is calculated by blending the second-order accurate central differencing scheme with a controlled amount of the first-order accurate upwind differencing scheme. Convection of k and ε was calculated using the upwind differencing scheme for all turbulent flow calculations.

Since mixed cell topologies are used for spatial discretisation, the colocated variable arrangement is the clear choice. The unphysical oscillations of pressure field associated with this arrangement are avoided by using the methodology of Rhie and Chow [1983].

7.2 SOLUTION ALGORITHM AND MULTIGRID ACCELERATION

The coupled systems of nonlinear algebraic equation resulting from discretisation of the governing equations, each pertaining to related flow variable, were solved in a sequential manner using the SIMPLE algorithm of Patankar and Spalding [1972]. A sequential or segregated approach was chosen because of the much smaller memory demands compared to simultaneous algorithms.

The SIMPLE algorithm sets a framework in which linearisation of discretised system of algebraic equations is achieved, and the SIMPLE (or outer) iterations are used to resolve non-linearities and inter-variable coupling. The linearisation facilitates the use of well-established solvers for the systems of linear algebraic equations.

The convergence of the SIMPLE algorithm was accelerated using Full Approximation Storage (FAS) multigrid method in a manner suggested by Sivaloganathan & Shaw [1988]. The algorithm is based on a fixed V-cycle and equipped with prolongation and restriction operators able to deal with multi-dimensional unstructured meshes made up of cells of arbitrary topologies.

The non-symmetric systems of linear algebraic equations resulting from discretisation of the momentum and turbulence model were solved using Gauss-Seidel iteration. The decision was made after several solvers, including Gauss-Seidel iteration, incomplete LU decomposition and the incomplete Cholesky preconditioned bi-conjugate gradient methods, were tested for the driven-cavity flow at $Re=100$. For all solvers, the number of fine-grid iterations remained the same irrespective of solver being used, indicating that the overall convergence rate is limited by the convergence properties of the base algorithm for interequation coupling and handling of non-linearities (i.e. SIMPLE) rather than on the solver used for solution of linearised equation sets. Since the Gauss-Seidel method required no additional memory, it was used for smoothing of the non-symmetric systems of linear equations. The incomplete Cholesky preconditioned Conjugate gradient method was used for the solution of the pressure-

correction equation, mainly because of its good convergence properties, even when the matrix does not exhibit diagonal dominance, which is the case with this equation.

The multigrid convergence behaviour, for the well-known 2D driven-cavity flow at Reynolds numbers of 100 and 1000, is in accordance with the theoretical expectations, in that an almost constant convergence rate is observed. The resulting savings in CPU time are up to two orders of magnitude at the finer mesh levels. The first-order prolongation operator, defined by expression (4.43), performs better than the zero-order one, reducing the total CPU time almost two times, in the calculation of the driven cavity flow at $Re=1000$. This result is in agreement with the conclusions of Thompson and Ferziger [1989]. Based on this numerical experiment the first-order prolongation operator was used for all other calculations.

Multigrid solvers deal effectively with algebraic systems describing elliptic problems. For the problems which do not have an elliptic nature or whose partial differential operator has a non-elliptic component, the performance of standard multigrid methodologies, like the one used in the present study, tends not to be the optimum one, due in part to the fact that the solutions become more complex.

The poor coarse-grid approximation of the fine-grid problem is one of the major causes of slow convergence of multigrid cycles. The poor approximation can be caused by local factors (e.g. treatment of the boundaries), but very often, and in particular in high-Reynolds flows, the coarse grid fails to approximate the fine-grid problem well enough throughout the domain. This problem is especially pronounced in calculations of turbulent flows and is believed to be one of the factors which caused FMG performance degradation in the present study.

The eddy viscosity depends on the mesh resolution quite considerably. For this reason, the effective viscosity distribution may look very different on the coarsest and the finest meshes in the early stages of the multigrid cycle. The resulting poor performance during this early stage can be critical, either hindering convergence altogether, or reducing the multigrid

efficiency considerably. A way of dealing with this problem adopted in this study is to obtain the field of eddy viscosity on the coarse meshes by interpolating the 'frozen' values from the finest one deactivating the coupling between the velocity field and the fields of k and ε on coarse grids. Although this helps to get or to improve convergence, it does not take the full advantage of the coarse-grid iteration, because in reality the eddy viscosity field develops as velocity field develops, both simultaneously affecting each other.

The use of turbulent wall functions raises additional problems, which can be important for multigrid efficiency. The non-linear nature of wall functions requires that a rather fine numerical resolution is needed before their spatial variation will appear to be 'smooth' on a grid. This means that for the implementation of wall functions as in the present study, the coarsest mesh used for multigrid, cannot be too coarse, otherwise the boundary conditions cannot respond 'accurately' to the changes of dependent variables in the near-wall region. If this is the case, since the whole solution is driven by boundary conditions, the representation of the fine-grid solution on the coarsest mesh may be poor, and then the coarse-grid corrections will not efficiently drive the fine-grid solution towards convergence.

It should however be stressed to those who do not seek the optimum performance that they can still benefit from the available multigrid algorithms. In particular these algorithms can offer considerable savings in CPU times, as can be seen from the results of this study.

Multigrid performed on five uniformly-refined grid levels reduced the CPU times almost 10 times for the case of flow through an orifice plate. To get this speed-up, the coarsest grid was not used for multigrid, for reasons explained above. To obtain a better speed-up factor for the central-differencing prediction, the calculations on the coarser grids were performed using the UD scheme, while the CD scheme was used on the finest grid. This practice resulted in some reduction in CPU time. The same practice was tried for the driven-cavity case, but here, poorer convergence rate was observed (CPU time has increased 10%-20%). Theoretical speculations suggest that the use of the higher-order

differencing scheme on the coarser grids leads to better multigrid convergence behaviour, since the coarse-grid equation better approximates fine-grid solution. This can be observed from the multigrid results for the driven-cavity flow at $Re=1000$, where 25 fine-grid iterations are needed on the 256×256 grid to get convergence using the CD scheme compared to 38 iteration using the UD scheme. However, the driven-cavity flow at this Reynolds number produces relatively small local Péclet numbers, even on the coarsest mesh, and the coarse-grid equation with the CD scheme gives a smooth solution correction field. For the orifice case this is not true, since on the coarse grid the CD scheme can produce first instability, which slows down the convergence rate on that grid, and second 'wiggly' corrections, which, when extrapolated to finer grids may introduce high-frequency error components requiring additional fine-grid iterations to reduce them.

Predictions for the turbulent flow over a prismatic obstacle and the swirling flow inside a gas-turbine combustor were done using the CD scheme on all grids. The FMG algorithm performance on four-uniformly refined grids is about 5 and 3.3 times faster than SGM for those two cases respectively. The modest speed-up for the combustor case could, among other things, be due to insufficient numerical resolution of the finest grid employed, such that solution was not sufficiently smooth on the finest and the representation of fine-grid solution on the coarse grids was therefore not good.

As the available computer power (including the computer memory) increases, flow solutions with small numerical error are becoming possible. However the CPU time necessary to obtain these solutions increases as the meshes become finer. The potential for significant CPU time reductions which lies in new algorithms cannot be overlooked. For all cases of this study, laminar or turbulent, the number of fine-grid iterations increases little as the mesh resolution increases, resulting in considerable CPU time savings. As efforts increase to obtain grid-independent solutions for turbulent cases, the absolute CPU time savings due to use of multigrid will be substantial. In these circumstances multigrid algorithms, even in their present state, have plenty to offer. In the present work no particular problems were encountered concerning the

stability or robustness of the FMG algorithm when applied to the cases which were presented. The main problem concerning the development of a computer program based on a multigrid algorithm, as the author can see it at present, is testing and maintaining such a code, since small inconsistencies, which can easily be introduced, are often manifested only by imperfect multigrid behaviour. There is always doubt if this imperfect behaviour is a result of bad implementation or if it has a more fundamental cause.

7.3 ERROR ESTIMATION AND ADAPTIVE MESH GENERATION

An analysis of the pressure coupling term appearing in the expression for the cell-face velocity was used in an original manner to estimate the truncation error which results from the second-order discretisation practice used to calculate the pressure gradient. The same expression was used to express this error in terms of velocity.

A novel error estimation technique, based on an estimation of truncation error, has been presented. The scheme is compact, since the truncation error estimate is assembled visiting only the immediate neighbours. This is a very useful concept when the scheme is to be applied to unstructured meshes made of cells of arbitrary topology because of the desired flexibility and simplicity. The estimation method can detect the errors present in the modelling of both convective and diffusive transport, using first- or second-order discretisation schemes. The methodology can also be extended to differencing schemes of higher order, but this will increase the complexity and computer cost. The technique does not take into account, in an explicit manner, the errors created in the discretisation of the cross-diffusion terms. However, this should not be a severe drawback, since cross-diffusion is not a dominant component in the case of large Péclet numbers or moderately non-orthogonal meshes.

The solution error may also be regarded as a transported variable, which is generated by the truncation error. The governing equations of

CFD are non-linear and coupled, and each of these equations has its corresponding error equation. These error equations make up a system of coupled non-linear equations, whose coefficients are based on the dependent variable fields. Because of the non-linear nature of those equations, it is very difficult to quantify the influence of one on another. Since there is uncertainty about truncation error estimation, it does not appear worthwhile to solve this system. Instead it is believed that the estimation of the maximum absolute error obtained from the novel truncation error estimation, as explained in Section 5.5 should be satisfactory for most of calculations.

Absolute error estimation based on Richardson extrapolation was used as well. In this method the solution error is deduced by comparing the solutions on coarse and finer meshes. Since the solution on the finer mesh advances towards a grid independent solution, the solution error estimation has information about the non-linear nature of the problem which is solved and about the transport properties of the solution error. Because of this, it will be useful to use this technique, when results on two or more grid levels are available.

Numerical tests were performed for the driven-cavity flow to evaluate quantitative and qualitative performance of the aforementioned error estimation techniques. Grid-independent solutions were obtained by systematically refining the grid. The velocity profiles in vertical and horizontal planes are reproduced using local refinement, with up to an order of magnitude less control volumes than in the case of uniformly-refined meshes. The use of local grid refinement and the FMG method reduces the CPU time between 75% for $Re=100$ and 88% for $Re=1000$, compared to the FMG without local refinement. The tau error estimations (Section 5.4) calculated from an error estimation based on Richardson extrapolation (Section 5.4.1) and estimating the truncation error in the novel manner described in Section 5.4.3 produced similar qualitative behaviour, leading to similar distributions of the control volumes in the locally-refined meshes.

The errors estimated by the novel technique show similar qualitative behaviour to the exact values. The maximum and averaged error

estimates were compared for the cavity flow at $Re=100$. The novel technique, as well as the one based on Richardson extrapolation underpredict the average solution error. The solution of the error transport equation produces better estimates of the average and maximum errors. This is to be expected, since the local solution error is not only affected by the presence of the local truncation error but also by the neighbouring error distribution.

A procedure has been devised to construct a hierarchy of numerical meshes to be used for multigrid method automatically and in an adaptive manner. The fine mesh is created from the coarser one by enriching numerical resolution in regions where the local error indicator is higher than a prescribed value.

In the present work the error is estimated for all dependant variables. This may not be necessary if one knows the variable which reflects the main properties of the flow field, as does pressure in the case of high Mach number flows. However, in general engineering applications, the safest way is to monitor the solution error in the fields of all dependant variables. Only this will guarantee that all processes relevant for the considered phenomena will be sufficiently resolved. This approach was adopted in the present study.

7.4 FUTURE RESEARCH

In order to improve the present solution procedure the following recommendations are made.

More work is required to establish reliable quantitative refinement criteria, although the techniques developed in this thesis provide a good starting point. Once this is done, automatic local refinement techniques, like the one presented in this thesis, will enable accurate solutions to be achieved more efficiently, since the safety margin which is added to the refinement criterion and is proportional to the uncertainty of the solution

error estimation, will be smaller, resulting in further memory and computer time reductions.

It would be worth devising techniques which could quantify the transport of the solution error in different regions of computational domain. This would enable one to suppress refinement in regions which are not important to the analysis, although they have large sources of solution error, as long as the latter is not transported into important regions. The error transport equation is a good starting point.

The use of wall functions does put limits on the mesh resolution in near-wall regions. To avoid this problem, it is clear that wall laws must be abandoned in favour of low-Reynolds-number turbulence models (e.g. Patel et al. [1985]), which permit integration down to the wall with no limits placed on the near-wall grid density. However, it should be noted that Lien [1992] found that the use of low-Re turbulence models resulted in a further deterioration of the MG speed-up ratios. One of the possible reasons identified for this was the high cell aspect ratios arising near the wall. Research into this important aspect of mesh refinement combined with multigrid techniques, would be welcome.

It would be useful to perform some further research into the poor multigrid convergence behaviour due to poor coarse-grid approximation of the fine-grid solution for high-Reynolds number flow calculations, especially for the turbulent flow.

So far multigrid acceleration techniques have not been extensively applied to the numerical simulation of many complex flow phenomena, such as turbulence, buoyancy, heat transfer, combustion and multiphase systems. Research and numerical experiments are necessary to obtain insight into the applicability and usefulness of multigrid methods for these classes of problems.

References :

- Altas, I. & Stephenson, J. W., 1991, "A two-dimensional adaptive mesh generation method", J. Comp. Phys., Vol. 94, p. 201.**
- Ames, W. F., 1979, Numerical methods for partial differential equations, Academic Press, New York.**
- Amsden, A. A., Ramshaw, J. D., O'Rourke, P. J. & Dukovicz, J. K., 1985, "KIVA: a computer program for two- and three-dimensional fluid flows with chemical reactions and fuel sprays.", Report LA 10245-MS, Los Alamos National Laboratory.**
- Aris, R., 1989, Vectors, tensors and the basic equations of fluid mechanics, Dover Publications, New York.**
- Barcus, M., Peric, M. & Scheuerer, G., 1987, "A control volume based full multigrid procedure for the prediction of two-dimensional, laminar, incompressible flows", Proc., 7th GAMM conference on Numerical Methods in Fluid Dynamics, Sept. 9-11, 1987, Louvain-la-Neuve, Belgium.**
- Becker, C. ,Ferziger, J. H., Peric, M., Scheuerer, G., 1988, "Finite volume multigrid solutions of the two-dimensional incompressible Navier-Stokes equations", Proc., 4th GAMM seminar, January 22-24, 1988, Kiel.**
- Berger, M. J., 1982, "Adaptive mesh refinement for hyperbolic partial differential equations", PhD Thesis, Department of Computer Science, Stanford University, Stanford, CA.**
- Berger, M. J. & Jameson, A., 1984, "Automatic adaptive grid refinement for the Euler equations", AIAA J., Vol. 23, p. 561.**

- Boris, J. P. & Book, D. L., 1973,** "Flux- corrected transport. I.SHASTA, a fluid transport algorithm that works", *J. Comp. Phys.*, Vol. 11, p. 38.
- Boussinesq, J., 1877,** "Essai sur la théorie des eaux courantes", *Mém. prés. Acad. Sci. XXIII*, 46, Paris.
- Brandt, A., 1977,** "Multi-level adaptive solutions to boundary-value problems", *Math. Comp.*, Vol. 31, p. 333.
- Brandt, A., 1980,** "Multilevel adaptive computations in fluid dynamics", *AIAA J.*, Vol. 18, p. 1165.
- Brandt, A., 1982,** *Lecture notes in mathematics*, Vol. 960, Springer-Verlag, Berlin, p. 220.
- Brandt, A. & Yavneh, I., 1991,** "Inadequacy of first-order upwind difference schemes for some recirculating flows", *J. Comp. Phys.*, Vol. 93, p. 128.
- Brandt, A. & Yavneh, I., 1992,** "On multigrid solution of high-Reynolds incompressible entering flows", *J. Comp. Phys.*, Vol. 101, p. 151.
- Brandt, A. & Yavneh, I., 1993,** "Accelerated multigrid convergence and high-Reynolds recirculating flows", *SIAM J. Sci. Comput.*, Vol. 14, No. 3, p. 607.
- Briggs, W. L., 1987,** *A multigrid tutorial*, SIAM, Philadelphia.
- Caretto, L. S., Curr, R. M. & Spalding, D. B., 1972a,** "Two numerical methods for three-dimensional boundary layers", *Comp. Meth. Appl. Mech. Eng.*, Vol. 1, p. 39.
- Caretto, L. S., Gosman, A. D., Patankar, S. V. & Spalding, D. B., 1972b,** "Two calculation procedures for steady, three-dimensional flows with recirculation", *Proc. 3rd Int. Conf. Num. Meth. Fluid Dynamics*, Paris, Vol. 2, p. 60.

- Caruso, S., 1985,** "Adaptive grid techniques for fluid flow problems", PhD Thesis, Thermosciences Division, Department of Mechanical Engineering, Stanford University, CA.
- Caughey, D. A. & Jameson, A., 1982,** "Basic advances in the finite volume method for transonic potential flow calculations." In: "Numerical and physical aspects of aerodynamic flows", ed. by T. Cebeci, Springer-Verlag, Berlin.
- Cebeci, T. & Smith, A. M. O., 1974,** Analysis of turbulent boundary layers, Academic Press, New York.
- Chapman, M., 1981,** "FRAM - Nonlinear damping algorithms for the continuity equation", J. Comp. Phys., Vol. 44, p. 84.
- Demirdzic, I. , Gosman, A. D. & Issa, R. I., 1980,** "A finite-volume method for the prediction of turbulent flow in arbitrary geometries", Proc. of Seventh Int. Conf. on Numerical Methods in Fluid Dynamics, Stanford, Springer-Verlag, Berlin, p. 144.
- Demirdzic, I., 1982,** "A finite volume method for computation of fluid flow in complex geometries", PhD Thesis, University of London.
- Demirdzic, I., 1991,** Private communication.
- Demkowicz, L., Devloo, P. & Oden, J. T., 1985,** "On an h-type mesh-refinement strategy based on minimisation of interpolation errors", Comp. Meth. Appl. Mech. and Eng., Vol. 53, p. 67.
- Denny, V. E. & Landis, R. B., 1972,** "A new method for solving two-point boundary value problems using optimal node distribution", J. Comp. Phys., Vol. 9, p. 120.
- Dupon, T., Kendall, R. & Rachford, H. H., 1968,** "An approximate factorization procedure for solving self-adjoint elliptic difference equations", SIAM J. Numerical Analysis, Vol. 5, p. 559.

- Fletcher, R., 1976,** "Conjugate gradient methods for indefinite systems", In: Numerical analysis, ed. by G. A. Watson, Lect. Notes in Math., 506, Springer-Verlag, Berlin, p. 227.
- Fletcher, C. A. J., 1991,** Computational techniques for fluid dynamics - Vol. 1: fundamental and general techniques, Springer-Verlag, Berlin.
- Fodorenko, R. P., 1962,** "A relaxation method for solving elliptic difference equations ", U. S. S. R. Computational Math. & Math. Phys., Vol. 1, p. 1092.
- Galpin, P. F. , Van Doormaal, J. P. & Raithby, G. D., 1985,** "Solution of the incompressible mass and momentum equations by application of a coupled equation linear solver", Int. J. Numer. Meth. Fluids, Vol. 5, p. 615.
- Garabedian, P., 1964,** Partial differential equations, John Wiley & Sons, New York.
- Gaskell, P. H., Lau, A. K. C. & Wright, N. G., 1987,** "Two efficient solution strategies for use with high-order discretisation schemes, in the simulation of fluid flow problems", 5th International Conference on Numerical Methods in Laminar and Turbulent Flow, Montreal, Quebec, Canada.
- Ghia, U., Ghia, K. N. & Shin, C. T., 1982,** "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method", J. Comp. Phys., Vol. 48, p. 387.
- Gosman, A. D., Pun, W. M., Runchal, A. K., Spalding, D. B. & Wolfstein, M., 1969,** Heat and mass transfer in recirculating flows, Academic Press, London.
- Gosman, A. D. & Lai, K. Y. M., 1982,** "Finite difference and other approximations for the transport and Navier-Stokes equations", IAHR Symposium on Refined Modelling of Flows, Paris.
- Gosman, A. D. & Ideriah, H. J. K., 1983,** "TEACH-2e: A general computer program for two-dimensional, turbulent, recirculating flows", Report,

Fluids Section, Department of Mechanical Engineering, Imperial College, London.

Gresho, P. M. & Lee, R. L., 1981, "Don't suppress the wiggles - they're telling you something !", *Computers and Fluids*, Vol. 9, p. 223.

Gustaffson, I., 1978, "A class of first order factorization methods", *BIT*, Vol. 18, p. 145.

Hackbusch, W., 1981, "Introduction to multigrid methods for the numerical solution of boundary value problems", in *Computational Fluid Dynamics*, Von Karman Institute for Fluid Dynamics Lecture Series 1981-5, Rhode Saint Genese, Belgium.

Hackbusch, W., 1985, "Multigrid methods and applications", *Computational mathematics*, Vol. 4, Springer-Verlag, Berlin.

Hageman, L. A. & Young, D. M., 1981, *Applied iterative methods*, Academic Press, New York.

Harten, A., 1984, "On a class of high resolution total variation stable finite difference schemes", *SIAM J. Numerical Analysis*, Vol. 21, p. 1.

Hawken, D. F., Gottlieb, J. J. & Hansen, J. S., 1991, "Review of some adaptive node-movement techniques in finite-element and finite-difference solutions of partial differential equations", *J. Comp. Phys.*, Vol. 95, p. 254.

Hestenes, M. R. & Stiefel, E., 1952, "Methods of conjugate gradients for solving linear systems", *Nat. Bur. Standards J. Res.*, Vol. 49, p. 409.

Hirsch, C., 1991a, *Numerical computation of internal and external flows* Vol. 1: fundamentals of numerical discretization, John Wiley & Sons, New York.

Hirsch, C., 1991b, *Numerical computation of internal and external flows* Vol. 2: computational methods for inviscid and viscous flows, John Wiley & Sons, New York.

- Hsu, C., 1981,** "A curvilinear-coordinate method for momentum, heat and mass transfer in domains of irregular geometry", PhD Thesis, University of Minnesota.
- Issa, R. I., 1986,** "Solution of the implicitly discretised fluid flow equations by operator-splitting", J. Comp. Phys., Vol. 62, p. 40.
- Jang, D. S., Jetli, R. & Acharya, S., 1986,** "Comparison of the PISO, SIMPLER and SIMPLEC algorithms for the treatment of the pressure-velocity coupling in steady flow problems". Num. Heat Transfer, Vol. 10, p. 209.
- Jayatillaka, C. L. V., 1969,** "The influence of Prandtl number and surface roughness on the resistance of the laminar sublayer to momentum and heat transfer". Report TWF/R/2, Mech. Eng. Dept., Imperial College.
- Jennings, A., 1977,** Matrix computation for engineers and scientists, John Wiley & Sons, New York.
- Jones, W. P. & Launder, B. E., 1972,** "The prediction of laminarisation with a two-equation model of turbulence", Int. J. Heat Mass Transfer, Vol. 15, p. 301.
- Kershaw, D. S., 1978,** "The incomplete Cholesky conjugate gradient method for the iterative solution of systems of linear equations", J. Comp. Phys., Vol. 26, p. 43.
- Khosla, P. K. & Rubin, S. G., 1974,** "A diagonally dominant second-order accurate implicit scheme", Comput. Fluids, Vol. 2, p. 207.
- Khosla, P. K. & Rubin, S. G., 1981,** "A conjugate gradient iterative method", Comput. Fluids, Vol. 9, p. 109.
- Korn, G. A. & Korn, T. M., 1968,** Mathematical handbook for scientists and engineers, McGraw-Hill, New York.
- Koutmos, P., 1985,** "An isothermal study of gas turbine flows", PhD Thesis, University of London.

- Lacroix, R., Camarero, R. & Tapucu, A., 1984,** "Multigrid scheme for thermohydraulic flow", *Num. Heat Transfer*, Vol. 7, p. 375.
- Lai, K. Y. M., 1982,** "Numerical analysis of fluid transport phenomena", PhD Thesis, University of London.
- Larousse, A., Martinuzzi, R. & Tropea, C., 1991,** "Flow around surface-mounted, three-dimensional obstacle", Eight Symposium on Turbulent Shear Flows, Technical University of Munich, September 9-11, 1991.
- Latimer, B. R. & Pollard, A., 1985,** "Comparison of pressure-velocity coupling solution algorithms", *Num. Heat Transfer*, Vol. 8, p. 635.
- Launder, B. E. & Spalding, D. B., 1974,** "The numerical computation of turbulent flows", *Comp. Meth. Appl. Mech. and Eng.*, Vol. 3, p. 269.
- Lax, P. D. & Wendroff, B., 1964,** "Difference schemes for hyperbolic equations with high order of accuracy", *Comm. Pure and Applied Mathematics*, Vol. 17, p. 381.
- Leschziner, M. A., 1980,** "Practical evaluation of three finite difference schemes for the computing of steady-state recirculating flows", *Comp. Meth. Appl. Mech. Eng.*, Vol. 23, p. 293.
- Lien, F. S. & Leschziner, M. A., 1991,** "Multigrid convergence acceleration for complex flow including turbulence", International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, Vol. 98, p. 277.
- Lien, F. S., 1992,** "Computational modelling of 3D flow in complex ducts and passages", PhD Thesis, University of Manchester.
- Lin, C. A. & Leschziner, M. A., 1989,** "Computation of three-dimensional injection into swirling combustor-model flow with second-moment closure", In: "Numerical Methods in Laminar and Turbulent Flow", Vol 6., Part 2., ed. by C. Taylor, P. Gresho R. L. Sani, J. Haüser, Pineridge Press, Swansea, p. 1711.

- Löhner, R., Morgan, K. & Zienkiewicz, O. C., 1985,** "An adaptive finite element procedure for compressible high speed flows", Comp. Meth. Appl. Mech. Eng., Vol. 51, p. 441.
- MacCormack, R. W., 1969,** "The effect of viscosity in hypervelocity impact cratering", AIAA Paper 69-354.
- Mavriplis, D. J., 1988,** "Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes", AIAA J., Vol. 26, p. 824.
- McCormick, S. F., 1987,** Multigrid methods, Vol. 3, SIAM Frontiers Series, SIAM, Philadelphia.
- McGuirk, J. J. & Rodi, W., 1978,** "A depth-averaged mathematical model for the near field of side discharges into open channel flow", J. Fluid Mechanics, Vol. 86, p. 761.
- McGuirk, J. J., Taylor, A. M. K. P. & Whitelaw, J. H., 1980,** "The assessment of numerical diffusion in the upwind-differencing calculations", Turbulent Shear Flow, Vol. 2, p. 206.
- Meijerink, J. A. & Van der Vorst, H. A., 1977,** "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M matrix", Math. Comp. , Vol. 31, p. 148.
- Meijerink, J. A. & Van der Vorst, H. A., 1981,** "Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems", J. Comp. Phys., Vol. 44, p. 134.
- Mulder, W. A., 1989,** "A new multigrid approach to convection problems", J. Comp. Phys., Vol. 83, p. 303.
- Nail, G. H., 1991,** "A study of 3-dimensional flow through orifice meters", PhD dissertation, Texas A&M University.
- Oden, J. T., Strouboulis, T. & Devloo, P., 1986,** "Adaptive finite element methods for the analysis of inviscid compressible flow: PART I. Fast refinement/unrefinement and moving mesh methods for

unstructured meshes", Comp. Meth. Appl. Mech. and Eng., Vol. 59, p. 327.

Oden, J. T., Demkowicz, L., Rachowicz, W. & Westermann, T. A., 1989, "Toward a universal h-p adaptive finite element strategy, PART 2. A posteriori error estimation, Comp. Meth. Appl. Mech. and Eng., Vol. 77, p. 113.

Palma, J. M. L. M., 1988, "Mixing in non-reacting gas turbine combustor flows", PhD Thesis, University of London.

Patankar, S. V. & Spalding, D. B., 1972, "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", Int. J. Heat Mass Transfer, Vol. 15, p. 1787.

Patankar, S. V., 1980, Numerical Heat Transfer and Fluid Flow, Hemisphere, Washington D C.

Patel, M., Cross, M. & Markatos, N. C., 1987, "An evaluation of eleven discretisation schemes for predicting elliptic flow and heat transfer in supersonic jets", Int. J. Heat Mass Transfer, Vol. 30, p. 1907.

Patel, V. C., Rodi, W. & Scheuerer, G., 1985, "Turbulence models for near-wall and low Reynolds number flows: a review", AIAA J., Vol. 23, p. 1308.

Peric, M., 1985, "A finite volume method for the prediction of three-dimensional flow in complex ducts", PhD Thesis, University of London.

Peric, M., Ruger, M. & Scheuerer, G., 1989, "A finite volume multigrid method for calculating turbulent flows", 7th Symp. Turbulent Shear Flows, Stanford University, p. 7.3.1.

Peric, M., 1990, Private communication.

Phillips, R. E., 1984, "A multilevel-multigrid method for recirculating flows", DPhil Thesis, Pennsylvania State University.

- Phillips, R. E. & Schmidt F. W., 1984,** "Multigrid techniques for the numerical solution of the diffusion equation", *Num. Heat Transfer*, Vol. 7, p. 251.
- Phillips, R. E., Miller, R. F. & Schmidt, F. W., 1985,** "A multilevel-multigrid algorithm for turbulent recirculating flows", *Proc. 5th Symp. on Turb. Shear Flows*, Stanford University, p. 20.21.
- Pope, S. B. & Whitelaw, J. H., 1976,** "The calculation of near-wake flows", *J. Fluid Mech.*, Vol. 73, p. 9.
- Prakash, C., 1981,** "A finite element method for predicting flow through ducts with arbitrary cross section", *PhD Thesis, University of Minnesota*.
- Raithby, G. D., 1976,** "Skew upstream differencing schemes for problems involving fluid flow", *Comp. Meth. Appl. Mech. Eng.*, Vol. 9, p. 151.
- Raithby, G. D. & Schneider, G. E., 1979,** "Numerical solution of problems in incompressible fluid flow: treatment of the velocity-pressure coupling", *Num. Heat Transfer*, Vol. 2, p. 417.
- Rhie, C. M., 1981,** "A numerical study of the flow past an isolated airfoil with separation", *PhD Thesis, University of Illinois, Urbana-Champaign*.
- Rhie, C. M. & Chow, W. L., 1983,** "A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation", *AIAA J.*, Vol. 21, p. 1525.
- Roache, P. J., 1972,** *Computational fluid dynamics*, Revised Printing, Hermossa Publisher, Albuquerque, New Mexico.
- Rubin, S. G. & Khosla, P. K., 1981,** "Navier-stokes calculations with a coupled strongly implicit method - I: finite difference solutions ", *Computers and Fluids*, Vol. 9, p. 163.
- Scarborough, J. B., 1958,** *Numerical mathematical analysis*, John Hopkins Press, Baltimore.

- Schlichting, H., 1968, Boundary layer theory, McGraw-Hill, New York**
- Scipio, L. A., 1967, Principles of continua with application, John Wiley & Sons, New York.**
- Sengupta, S., Häuser, J., Ekermann, P. R. & Thompson, J. F., 1988, Numerical grid generation in computational fluid mechanics '88, Pineridge Press, Swansea.**
- Sivaloganathan, S. & Shaw, G. J., 1988, "A multigrid method for recirculating flow", Int. J. Num. Meth. Fluids, Vol. 8, p. 417.**
- Slattery, J. C., 1981, Momentum, energy and mass transfer in continua, Robert E. Krieger Publishing Company, Huntington, New York.**
- Smith, D. M., 1990, "Fluid dynamic calculations using the multigrid approach", PhD Thesis, University of London.**
- Sonneveld, P., Wesseling, P. & de Zeeuw, P. M., 1985, "Multigrid and conjugate methods as convergence acceleration techniques", In: "Multigrid methods for integral and differential equations", ed. by D. J. Paddon & H. Holstein, The institute of mathematics & its applications conference series, 3, Clarendon press, Oxford, p. 117.**
- Spalding, D. B., 1972, "A novel finite-difference formulation for differential expressions involving both first and second derivatives", Int. J. Num. Meth. Eng., Vol. 4, p. 551.**
- STAR-CD Version 2.1 - Manuals, 1991, Computational Dynamics Limited.**
- Stone, H. L., 1968, "Iterative solutions of implicit approximations of multi-dimensional partial differential equations", SIAM J. Numerical Analysis, Vol. 5, p. 530.**
- Stuben, K. & Trottenberg, U., 1982, "Multigrid methods, fundamental algorithms, model problem analysis and applications", Multigrid Methods, Lecture Notes in Mathematics, 960, p. 1, Springer-Verlag, Berlin.**

- Syed, S., Gosman, A. D. & Peric, M., 1985,** "Assessment of discretization schemes to reduce numerical diffusion in the calculation of complex flows", AIAA-85-0441.
- Tennekes, H. & Lumley, J. L., 1989,** A first course in turbulence, The MIT Press, Cambridge, Massachusetts.
- Thompson, J. F., 1984,** "Grid generation techniques in computational fluid dynamics", AIAA J., Vol. 22, p. 1505.
- Thompson, M. C. & J. H. Ferziger, 1989,** "An adaptive multigrid technique for the incompressible Navier-Stokes equations", J. Comp. Phys., Vol. 82, p. 94.
- Van der Vorst, H. A. & Sonneveld, P., 1990,** "CGSTAB, a more smoothly converging variant of CGS", Tech. Rep. 90-50, Delft University of Technology.
- Van Doormaal, J. P. & Raithby, G. D., 1985,** "An evaluation of the segregated approach for predicting incompressible fluid flows", National Heat Transfer Conference Denver, Colorado - August 4-7, 1985.
- Van Leer, B., 1982,** "Flux vector splitting for the Euler equations", In Proc. 8th International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, Berlin.
- Vanka, S. P., 1986,** "Block-implicit multigrid solution of Navier-Stokes equations in primitive variables", J. Comp. Phys. , Vol. 65, p. 138.
- Varga, R. S., 1962,** Matrix iterative analysis, Prentice-Hall, Englewood Cliffs, New Jersey.
- Vasilic-Melling, D., 1976,** "Three-dimensional turbulent flow past rectangular bluff bodies", PhD Thesis, University of London.
- Vinsome, P. K. W., 1976,** "ORTHOMIN, an iterative method for solving sparse sets of simultaneous linear equations", Society of Petroleum Engineers of AIME, paper SPE 5729.

Zedan, M. & Schneider, G. E., 1983, "Investigation of the simultaneous variable solution for velocity and pressure in incompressible fluid flow problems", AIAA-83-1519.

Zienkiewicz, O. C., 1977, The finite element method, McGraw-Hill, New York.

Ziman, H. J., 1990, "A computer prediction of chemically reacting flows in stirred tanks", PhD Thesis, University of London.

