MULTIGRID SOLUTIONS OF ELLIPTIC FLUID FLOW PROBLEMS

by

Nigel George Wright

Submitted in accordance with the requirements

for the Degree of Doctor of Philosophy

Department of Mechanical Engineering

University of Leeds

Leeds LS2 9JT

United Kingdom

September 1988

Acknowledgements

I would like to record my thanks to Dr. P.H. Gaskell for his invaluable advice and encouragement over the last three years of my work on this thesis, Prof. D.G. Crighton (DAMTP, University of Cambridge) for his constructive comments throughout this work and his reading of the final draft, and Prof. D. Bradley,F.R.S., for his support of this work.

I am grateful to Dawn Fuller for typing large parts of this thesis and for her companionship and support, especially in the difficult final phases of the work.

Thanks too, to my parents and sister who have supported and encouraged me throughout my education.

In addition I would like to take this opportunity to thank the staff and postgraduates in the Departments of Applied Mathematical Studies and Mechanical Engineering. They provided a stimulating environment in which to work and contributed to many useful discussions. Particular thanks must go to those on Level 11 who have made the undertaking of this work so enjoyable and to Dr. A.K.C. Lau for his advice.

Many personal friends have helped me whilst in Leeds, but particular thanks goes to Richard Everson, Sandra & DerrickFuller, Sue Golay, Rick Lavin, Chris Whitlow and the various residents of 5 Archery Street.

Financial support provided by the Science and Engineering Research Council and Rolls-Royce plc is gratefully acknowledged.

Publications

Gaskell, P.H. and Wright, N.G.

"A Multigrid Algorithm for the Investigation of Thermal Recirculating Fluid Flow Problems", Proceedings of the 5th International Conference on Numerical Methods for Thermal Problems, Part 2, ed. Lewis and Morgan, pp. 1202-1215 (1987).

"Multigrids Applied to a Solution Technique for Recirculating Fluid Flow Problems", Simulation and Optimisation of Large Systems, ed. A.J. Osiadacz, IMA Conference Series, Clarendon Press, Oxford, pp. 51-65 (1988).

"An Improved Multigrid Strategy for the Investigation of Highly Recirculating Fluid Flows", Proceedings of the AIAA/ASME/SIAM/APS 1st National Fluids Dynamics Congress, Part 1, Paper Number 88-3651, pp. 264-272 (1988).

Gaskell, P.H., Lau, A.K.C. and Wright, N.G.

"Two Efficient Solution Strategies for Use with High-Order Discretisation Schemes, in the Simulation of Fluid Flow Problems", Proceedings of 5th International Conference on Numerical Methods in Laminar and Turbulent Flows, Part 1, eds. Taylor, Habashi and Hafez, pp. 210-220 (1987).

"Comparison of Two Solution Strategies for Use with Higher Order Discretisation Schemes in Fluid Flow Simulation", International Journal for Numerical Methods in Fluids (Invited Paper) - in press.

"The Accurate Approximation and Economic Solution of Steady-State Convection Dominated Flows", Numerical Methods for Fluid Dynamics, ed. K.W. Morton and M.J. Baines, IMA Conference Series, Clarendon Press, Oxford - in press.

Abstract

An efficient FAS multigrid solution strategy is presented for the accurate and economic simulation of convection dominated flows. The use of a high-order approximation to the convective transport terms found in the governing equations of motion has been investigated in conjunction with an unsegregated smoothing technique.

Results are presented for a sequence of problems of increasing complexity requiring that careful attention be directed towards the proper treatment of different types of boundary condition. The classical two-dimensional problem of flow in a lid-driven cavity is investigated in depth for flows at Reynolds numbers of 100, 400 and 1 000. This gives an extremely good indication of the power of a multigrid approach.

Next, the solution methodology is applied to flow in a three-dimensional lid-driven cavity at different Reynolds numbers, with cross-reference being made to predictions obtained in the corresponding two-dimensional simulations, and to the flow over a step discontinuity in the case of an abruptly expanding channel. Although, at first sight, these problems appear to require only minor extensions to the existing approach, it is found that they are rather more idiosyncratic.

Finally, the governing equations and numerical algorithm are extended to encompass the treatment of thermally driven flows. The solution to two such problems is presented and compared with corresponding results obtained by traditional methods.

Nomenclature

The following is a list of frequently used symbols in this thesis. Additional notation is used and this is defined when introduced.

ϕ	A general scalar quantity
t	Time
x_j	The distance in the jth co-ordinate direction
Γ	Diffusion coefficient
S_ϕ	Source term for ϕ
u	Velocity in the x co-ordinate direction
v	Velocity in the y co-ordinate direction
w	Velocity in the z co-ordinate direction
h	Finite difference mesh spacing
T_{ij}	Truncation error at the point (x_i, y_j)
$\hat{L}_{ij}$	Differential operator
L_{ij}	Difference operator
ϕ_k	Normalised face value
Pe	Peclet number, $\frac{uh}{\Gamma}$
p	Pressure
ρ	Density
μ	Viscosity
ν	Kinematic viscosity
β	Exponential factor of increase in computer time with the number nodes
γ	Rate of residual reduction
λ	Wavelength
r_{ij}^c	Residual in the continuity equation
r_{ij}^u	Residual in the u-momentum equation
r_{ij}^v	Residual in the v-momentum equation

r_{ij}^t	Residual in the temperature equation
A	Coefficient in finite difference equation
ψ	Streamfunction
ω	Vorticity
Q^k	Solution vector on grid k
r^k	Residual vector on grid k
f^k	Constant source vector on grid k
L^k	Finite difference operator on grid k
q^k	Approximation vector for Q^k
s^k	Correction to q^k
F^k	Source vector for multigrid equations
M	Finest grid level
I_k^{k-1}	Interpolation operator from k to k-1
I_{k-1}^k	Interpolation operator from k-1 to k
η	Reduction factor in multigrid algorithm
FGWU	Fine grid work units
R^n	Residual at nth iteration
θ	Multigrid convergence factor
ϕ_N	ϕ at a boundary
Pr	Prandtl number, defined in text
Gr	Grashof number, defined in text
Ra	Rayleigh number, defined in text
Nu	Nusselt number, defined in text

Contents

Chapter 1

INTRODUCTION

1.1. Overview

In recent years high speed digital computers have begun to play an increasing role in engineering design. The field of Computational Fluid Dynamics (CFD) has evolved at an equally rapid rate such that the two together are now beginning to challenge the superiority of experimentation as a design tool capable of predicting the flow within, through and around complex engineering configurations. At present they represent complementary rather than competitive approaches, but the relative role of these two flow visualisation techniques is gradually changing. The time is not too far off when experimentation will be relegated to the secondary design phase, one of validation and refinement of global simulations for the entire flow field rather than for extensive parameter studies as in the past.

Computational Fluid Dynamics, although a relatively young discipline, covers a vast spectrum of interests and methodologies several of which have become research topics in their own right - grid generation techniques[47], body fitted coordinates[9], turbulence modelling[31], solution procedures[35,24] etc.. It is impossible therefore to give an exhaustive review of the subject; we can only skim the surface, taking care to place the work reported here into context.

Numerical simulation is well suited to the analysis of a wide range of complex fluid flows. In some cases it even permits investigations to be carried out for situations that cannot be readily or easily duplicated experimentally. For example, consider the flow of a coolant through a complicated pipe network adjacent to the core of a nuclear reactor. Such simulations are extremely useful in assisting an investigator to visualise how shape changes effect the global characteristics of a flow field. Prediction of the essential features of the flow pattern (such as recirculation zones) in such situations is important in order to enable engineers to design efficient and safe devices at minimum cost.

The discrete forms used to compute fluid flow are derived from approximations to the full Navier-Stokes equations[1] - a set of non-linear partial differential equations

which govern such motions. They comprise, in the main, finite element[36] and finite difference/volume techniques[38]. In each case the domain of interest is subdivided to produce a computational grid or mesh with appropriate boundary conditions at the periphery. The question as to which approach is best is somewhat academic since they both excel and fail under different circumstances. However, in line with general current engineering practise a finite volume formulation has been adopted here; primarily because it gives one the ability to solve large problems at a small computational cost, thus having the edge over a finite element approach which has the flexibility to handle rather more complex geometries.

The approaches presently available for dealing with the equations governing the motion of a fluid are hierarchical and can be categorised according to the degree of approximation involved :

(a) Solution of the full Navier-Stokes equations

(b) Solution of time-averaged Navier-Stokes equations

(c) Laminar viscous flow simulations

(d) Non-linear inviscid flow solutions

(e) Linearised inviscid flow solutions

Category (a) and (b) type solutions allow for the presence of turbulence in the system. The former is known as the Direct (or Large Eddy) Simulation Approach[37] - research in this area is quite intensive but solutions of this type are practically non-existent. Some very simple problems have been investigated but the computing power required to solve flows of engineering interest does not exist. The latter approach is realisable[27,26] and has therefore proved popular within the engineering community. It continues to be the focus for extensive research and development[25]. However, the principle difficulty that still has to be overcome is the realisation of a suitably general turbulence model.

We turn now to inviscid flow methods, categories (d) and (e). The former involve the solution of the Euler equations[1] and are now more-or-less established as accurate design tools within the aircraft industry, as a means of predicting the flow around a

class of aircraft components. Similarly, the category (e) approach is used routinely in aircraft and vehicle design[22] and is at a mature stage of development - panel and vortex lattice[20] methods belong to this set.

The incompressible steady-state flow problems to be considered here fall into category (c) and are governed by a set of elliptic, coupled partial differential equations which, when written in a discrete form give rise to a non-linear coupled matrix system, the solution vector of which is required to be found accurately and efficiently. How effectively this can be done reflects the main theme of this thesis.

1.2. Background

Having identified the problem to be addressed the strategy required to solve it accurately and efficiently can be formulated in terms of the discretisation employed on a given mesh and the method of solution of the resultant algebraic system of equations ,respectively.

For convection dominated flows the method of approximating the first order convective terms present in the conservation equations is of particular importance, having been the subject of controversy for a number of years. Indeed, it has spawned a whole series of publications on the subject[19,21,42,44,29].

It was soon recognised that the use of central differencing for the convection term produces unphysical oscillations in a solution causing it to diverge or at best be seriously corrupted[48,34]. Upwind differencing on the other hand, although stable, is beset with the problem of inherent false, or numerical, diffusion[38]. In 1972 Spalding[46] hit upon the idea of merging the two to produce what is now commonly know as the hybrid scheme. Unfortunately, this however does not solve the problem - switching between the two types of differencing is constrained in such a way that as the non-linearity of the flow increases the upwind approximation is used predominantly. Even so, it is worth noting at this point that the hybrid scheme still enjoys wide-spread usage in many off-the-shelf fluid flow software packages available to industry.

In 1979 the work of Leonard[28] ushered in a new realisation with regard to the inadequacy of the hybrid scheme. He proposed a new non-diffusive high-order approximations to convective transport - Quadratic Upstream Interpolation for Convection Kinematics (QUICK). The superiority of QUICK in relation to the solution of laminar flow problems has been reported by several authors[16, 19], the only noticeable drawback being associted with a slight increases in computing time. However since one is able to generate results on a course grid with QUICK that are much more accurate than those obtained on a very fine grid with hybrid differencing the increase incurred is more than compensated for. Unfortunately, the same cannot be said of QUICK for the case of turbulent flow simulation[14, 19] - the inherent lack of boundedness associated with this scheme can lead to disastrous consequences. For example, negative turbulent kinetic energies may arise which, besides being unphysical, may destroy the solution completely. Although we are not concerned in this thesis with simulating turbulent flow one would hope that the methodology presented may eventually be extended to such problems, in which case QUICK may be deemed to lack the element of robustness that one desires.

There has been a spate of activity of late to develop an accurate, high-order, bounded approximation to convective transport. Several such have been proposed[39, 40, 52], but they are all rather similar relying on rather ad hoc means for maintaining boundedness, which incidently cannot be guaranteed. However, a new scheme, Curvature Compensated Convective Transport (CCCT) developed by Gaskell and Lau[15] possess all of the above properties and guarantees boundedness. Also, it is very robust having been applied to a range of complex turbulent flows in various geometries[13], incorporating in some cases combustion effects[3]. This scheme exhibits another rather interesting feature in that both of the high order approximations mentioned above (central, QUICK) and others, can be obtained from its generic form, thus removing the need to code each one of them separately should the need arise. This then was the approximation adopted to model the convective transport terms contained in the governing equations of motion.

The task at hand is to develop a strategy for solving the large system of algebraic equations, generated by the discretisation process, as efficiently as possible. Here we have approached this problem with the aid of a multigrid technique which in itself raises several important considerations. In particular, what form of solver (smoother) to use? Note that when using a multigrid method the tendency is to use the term smoother rather than solver since the object is to smooth the error on a given grid rather than to solve the problem there exactly . The choice of smoother can be critical in relation to the performance of the multigrid algorithm.

The principle difficulty in relation to solving fluid flow problems numerically is that the pressure field, which drives the motion, is not known a priori. The classical way of overcoming this problem is to use a vorticity-streamfunction formulation whereby the explicit appearance of the pressure is eliminated from the transport equations[17]. However, the short comings of this approach are the difficulty of specifying boundary conditions for the vorticity and extension of the method to three-dimensions. Once again these problems are exacerbated in turbulent flow situations.

In recent years it has become common practise to adopt the primative variable formulation with regard to engineering problems; the velocity components and pressure (or pressure correction) being determined from their own transport equations. A literature search soon reveals the Semi-Implicit Pressure-Linked Equation (SIMPLE) algorithm attributed to Patankar and Spalding[33] to be, by far, the most popular method of solution in this case. Like the hybrid scheme it enjoys wide spread usage, being the principle methodology for most, if not all, commercially available software packages. Over the years variants of SIMPLE have emerged - SIMPLER[32], SIMPLEC[11] and PISO[23] but it is debatable as to whether they represent any great improvement over the original formulation when used in connection with complex flow situations. In all cases the matrix solver used in conjunction with this approach is the well known Tri-Diagonal Matrix Algorithm (TDMA)[7].

The SIMPLE algorithm is used extensively by both scientists and engineers alike which would suggest that it cannot be ignored as a possible smoother for use with a

multigrid algorithm. Clearly it would be preferable from the point of view of existing codes if an appropriate version of the latter could simply be "boot strapped" to them. Indeed, it is the fact that SIMPLE is at the heart of the Rolls-Royce PACE program that prompted Shaw and Sivaloganathan to investigate its suitability in such a role. They have shown SIMPLE to have good theoretical smoothing properties[43] and have obtained solutions to the well known lid-driven cavity problem on grids as fine as 1/64[45] - they were unable to go further because of computer storage limitations. Lonsdale[30] has also been successful in using a multigridded version of the SIMPLEC algorithm to investigate the problem of the flow of air between two rotating discs - the finest grid employed being 1/65.

It is important to stress however, that SIMPLE represents a segregated (or decoupled) solver. That is, the velocity and pressure fields are decoupled and solved sequentially, the latter being determined via a derived pressure equation. Another solver which falls into this category is the Distributive Gauss-Seidel (DGS)[6] approach but it has enjoyed only limited application. Obviously it would be preferable to solve the equations directly, thus requiring no iterative procedure. But this is impractical from both computer processing time and storage capacity requirements. Recently however, several attempts have been made to devise a solver that treats the variables simultaneously, in an unsegregated fashion, thus maintaining the physical coupling between them.

In 1983 Zedan and Schneider[50] proposed an unsegregated solution technique know as the Direct Banded Simultaneous Variable Solution (DBSVS) method. Applying it to a simple test problem in order to examine pressure-velocity coupling, the found it to be strongest in the immediate vicinity of the current node. This enabled them to devise a variant of the above called the Strongly Implicit Simultaneous Variable Solution (SISVS) method[51]. They reported some success in applying it to the lid-driven cavity problem at various Reynolds numbers. Unfortunately they were restricted to using a very course mesh, 1/10, and although the results look promising it is difficult to do a direct comparison with similar results obtained by other means. At

present, further applications of the method would appear to be restricted by the amount of storage required to accommodate the resultant coefficient matrices.

Similarly, the Coupled Equation Line Solver (CELS) proposed by Galpin, van Doormaal and Raithby in 1985[12], solves the conservation equations in their original form. The method proceeds line-by-line, the solution domain being swept in each co-ordinate direction separately until convergence is obtained. Unfortunately it requires special adaptations in order to maintain stability. This together with its rather complex overall nature effectively rules it out at present as an appropriate solver for use with a multigrid scheme.

The Block Implicit Method (BIM) on the other hand, suggested by Vanka[49], is easily comprehended and efficient. It is a point-by-point method in that each of the control volumes covering the solution domain are visited in turn, the velocities and pressures their being updated simultaneously. Consequently, each velocity is incremented twice thus ensuring the stability lacking with a single update. The methods simplicity and low operational count made it the obvious choice for use with, and in the development of, a multigrid method. Unlike the SIMPLE method it has not been possible to look at the theoretical smoothing properties of the BIM hence an intuitive approach had to be adopted in performing the computations presented in this thesis. It was gratifying therefore to have heard quite recently, from an independent source[41], that a preliminary, all be it limited, analysis has revealed the latter to have a superior smoothing rate.

There are two schools thought governing the use of multigrids. One concentrates on rigorous mathematical analysis of the convergence of multigrid algebraic solvers[18], the other addresses the practical development of efficient multigrid algorithms[4,5]. In terms of practical computational fluid dynamics (CFD) applications it is often only the latter approach that can be realistically pursued. Brandt started the wheels rolling in the seventies when he identified the practical significance of using multigrids and was the first to apply the principle to CFD problems. However his willingness to proceed without adherence to strict mathematical proof has been blamed for not drawing the

attention of numerical analysts' to the idea quickly enough. Similarly, in the early stages, he was unable to convince CFD experts of the value of multigrids. The breakthrough has only come during the last five years or so and today physicists, engineers and CFD experts alike are actively engaged in the multigrid area on a broad front.

Putting aside the question of mathematical rigour in relation to the application of multigrids there are two ways one might wish to proceed.

It is often the case in CFD situations that a large code already exists, for the solution of a particular class of fluid flow problems, that has taken many people a large number of years to develop. Such codes often take several minutes if not hours to run and any reductions that could be achieved via some form of multigrid approach would be welcome. Since the method of discretisation, the grid structure and the solution strategy have already be chosen ones only hope, from a cost effective point of view, is to introduce an element multigridding into the overall algorithm, as a means of accelerating its convergence rate, without having to completely rewrite the source code. Clearly, there are limitations as to what can be achieved by proceeding in this manner.

However, if one is starting from scratch there is a great deal more that can be done. The concern here is not merely to accelerate already existing convergence rates but to achieve optimal multigrid convergence for a given problem by designing a complete multigrid solution strategy. This is done first by selecting a suitable discretisation procedure and smoothing technique as outlined above, followed by adopting optimal grid structures and by optimal tailoring of the multigrid components to the flows under investigation.

1.3. Outline of Present Work

We begin by considering some important aspects of numerical approximations to the equations governing incompressible fluid flow; in particular the factors which dictate the choice of a suitable model for convective transport.

Different types of solver are reviewed in Chapter 3 in relation to their use with a multigrid algorithm. The Block Implicit Method is identified as being the most suitable and is applied to the classical fluids problem of flow in a two-dimensional lid-driven cavity[8], at three different Reynolds numbers. An optimal method of sweeping through the computational grid is formulated and the predicted flow fields obtained with both hybrid and CCCT differencing are compared.

In Chapter 4 a multigrid solution strategy is presented following a review of such practises in the past. The finer points of restriction and prolongation operators are discussed and the concept of fine grid work units is introduced. The multigrid version of the Block Implicit Method which results is applied once again to the two-dimensional lid-driven cavity problem. The information generated gives a clear insight into the advantages of using a multigrid approach to solve this problem. Grid independent convergence is achieved and savings in computer time in the region of two orders of magnitude are achieved when results are compared with the data given in Chapter 3 for the same flows.

The multigrid solution strategy described in Chapter 4 is extended in Chapter 5 to encompass a more complicated class of flow problems. Here careful consideration has to be given to the treatment of derivative boundary conditions and the associated transferral of information between grids. First, the flow in a three-dimensional lid-driven cavity is investigated at Reynolds numbers of 100 and 1000. Cross reference is made to their corresponding two-dimensional flows and once again the property of grid independent convergence is achieved by the multigrid algorithm. Comparisons are also shown for results obtained for the same problem using a straight forward SIMPLE approach[16]. Second, The flow through a two-dimensional sudden expansion is considered. This is an intrinsically more complicated problem because it contains an outflow derivative boundary condition. The handling of such a condition is not trivial. The smoothing method and the problem formulation used here were such that solutions could only be obtained at low Reynolds numbers.

In Chapter 6 the methodology is extended still further to look at two thermal problems - a square cavity with conducting walls[2] and one with insulated walls. The latter is perhaps more commonly known as the double-glazing problem[10]. The results obtained in both cases are extremely encouraging. Grid independent convergence is achieved in a fraction of the times reported by other author using traditional approachs to solve such problems.

Finally, conclusions and suggestions for future development of the present work are presented in Chapter 7.

References

1. Batchelor, G.K., *An Introduction to Fluid Dynamics*, CUP (1967).

2. Boonkkamp, J.H.M. ten Thije, *The Odd-Even Hopscotch Pressure Correction Scheme for the Computation of Free Convection in a Square Cavity*, Centre for Mathematics and Computer Science (1980).

3. Bradley, D., Gaskell, P.H., Lau, A.K.C., Missaghi, M., and Chin, S.B., ''Mathematical Modelling of Turbulent Combustion'', *Institute of Mechanical Engineers Conference on Engine Combustion*, (1987).

4. Brandt, A., ''Multi-level Adaptive Solutions to Boundary-Value Problems'', *Mathematics of Computation* **31**(138) pp. 333-390 (April 1977).

5. Brandt, A., ''Multilevel Adaptive Computation in Fluid Dynamics'', *AIAA Journal* **18** pp. 1165-1172 (Oct. 1980).

6. Brandt, A. and Dinar, N., ''Multigrid Solution to Elliptic Flow Problems'', pp. 53-147 in *Numerical Methods in Partial Differential Equations*, ed. S. Parter, (1977).

7. Burden, R.L, Faires, J.D., and Reynolds, A.C., *Numerical Analysis*, Prindle, Weber and Schmidt, Boston, Massachussetts (1981).

8. Burggraf, O.R., ''Analytical and Numerical Studies of The Structure of Steady Separated Flows'', *Journal of Fluid Mechanics* **2**(1) pp. 113-151 (1966).

9. Burns, A.D., Wilkes, N.S., Jones, I.P., and Kightley, J.R., *FLOW3D: Body-Fitted Coordinates, AERE-R 12262,* Computer Science and Systems Division, Harewell Laboratory (1986).

10. Davis, G. de Vahl, ''Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution'', *International Journal for Numerical Methods in Fluids* **3** pp. 249-264 (1983).

11. Doormaal, J.P. Van and Raithby, G.D., ''Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows'', *Numerical Heat Transfer* **7** pp. 147-163 (1984).

12. Galpin, P.F., Doormaal, J.P. Van, and Raithby, G.D., ''Solution of the Incompressible Mass and Momentum Equation by Application of a Coupled Equation Line Solver'', *International Journal for Numerical Methods in Fluids* **5** pp. 615-625 (1985).

13. Gaskell, P.H. and Lau, A.K.C., ''An Assessment of Direct Stress Modelling for Elliptic Turbulent Flows with the Aid of a Non-Diffusive Boundedness Preserving Discretisation Scheme'', pp. 351-362 in *Proceedings of The 5th International Conference on Numerical Methods in Laminar and Turbulent Flow, Part 1*, ed. Taylor, Habashi and Hafez, (1987).

14. Gaskell, P.H. and Lau, A.K.C., ''The Method of Curvature Compensation and its Use in the Prediction of Highly Recirculating Flows'', pp. 272-279 in *Proceedings of the AIAA/ASME/SIAM/APS 1st National Fluid Dynamics Congress, Part 1,* (1988).

15. Gaskell, P.H. and Lau, A.K.C., ''Curvature Compensated Convective Transport : SMART, a New Boundedness Preserving Transport Algorithm'', *International Journal for Numerical Methods in Fluids* **8** pp. 617-641 (1988).

16. Gaskell, P.H., Lau, A.K.C., and Wright, N.G., ''Comparison of Two Strategies for Use with Higher Order Discretisation Schemes in Fluid Flow Simulation'', *International Journal for Numerical Methods in Fluids - in press,* ().

17. Ghia, U., Ghia, K.N., and Shin, C.T., "High Re Solutions for Incompressible Flow Using the Navier- Stokes Equations and a Multigrid Method", *Journal of Computational Physics* **43**(3)(December 1982).

18. Hackbusch, W., *Multi-grid Methods and Applications*1985.

19. Han, T., Humphrey, J.A.C., and Launder, B.E., "A Comparison of Hybrid and Quadratic-Upstream Differencing in High Reynolds Number Elliptic Flows", *Computer Methods in Applied Mechanics and Engineering* **29** pp. 81-95 (1981).

20. Hess, J.L. and Smith, A.M.O., "Calculation of Potential Flow about Arbitrary Bodies", *Progress in Aeronautical Sciences* **8** pp. 1-138 (1967).

21. Huang, P.G., Launder, B.E., and Leschziner, M.A., "Discretisation of Non-linear Convection Processes : A Broad-Range comparison of Four Schemes", *Computer Methods in Applied Mechanics and Engineering* **48** pp. 1-24 (1985).

22. Hucho, W.H., *Aerodynamics of Road Vehicles*, Butterworths (1987).

23. Issa, R.I., Gosman, A.D., and Watkins, A.P., "The Computation of Compressible and Incompressible Recirculating Flows by a Non-Iterative Implicit Scheme", *Journal of Computational Physics* **62** pp. 66-82 (1986).

24. Jang, D.S., Jetli, R., and Acharya, S., "Comparison of the PISO, SIMPLER and SIMPLEC Algorithms for the Treatment of the Pressure-Velocity Coupling in Steady Flow Problems", *Numerical Heat Transfer* **10** pp. 209-228 (1986).

25. Lau, A.K.C., "Mathematical Modelling of Non-Reacting and Reacting Recirculation Flows", *Ph.D. thesis*, University of Leeds, (1987).

26. Launder, B.E., Reece, G.J., and Rodi, W., "Progress in the Development of a Reynolds-Stress Turbulence Closure", *Journal of Fluid Mechanics* **68** pp. 537-566 (1975).

27. Launder, B.E. and Spalding, D.B., "The Numerical Computation of Turbulent Flows", *Computer Methods in Applied Mechanics and Engineering* **3** pp. 269-289 (1974).

28. Leonard, B.P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", *Computational Methods in Applied Mechanics and Engineering* **19** pp. 59-98 (1979).

29. Leschziner, M.A., "Practical Evaluation of Three Finite Difference Schemes for the Computation of Steady State Recirculating Flows", *Computational Method in Applied Mechanics and Engineering* **23** pp. 293-312 (1980).

30. Lonsdale, G., "Solution of a Rotating Navier-Stokes Problem by a Nonlinear Multigrid Algorithm", *International Journal for Numerical Methods in Fluids*, (1988).

31. Nallasamy, M., "Turbulence Models and their Application to the Prediction of Industrial Flows: A Review", *Computers and Fluids* **15**(2) pp. 151-194 (1987).

32. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing (1980.).

33. Patankar, S.V. and Spalding, D.B., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows", *International Journal of Heat Mass Transfer* **15** pp. 1787-1806 (1972).

34. Patel, M.K., Markatos, N.C., and Cross, M., "A Critical Evaluation of Seven Discretisation Schemes for Convection-Diffusion Equation", *International Journal for Numerical Methods in Fluids* **5** pp. 225-244 (1985).

35. Raithby, G.D. and Schneider, G.E., "Numerical Solution of Problems in Incompressible Fluid Flow: Treatment of the Velocity-Pressure Coupling", *Numerical Heat Transfer* **2** pp. 417-440 (1979).

36. Rao, S.S., *The Finite Element Method in Engineering*, Pergamon Press (1982).

37. Riley, J.J. and R.W.Metcalfe,, "Direct Numerical Simulations of the Turbulent Wake of an Asymmetric Body", pp. 78-93 in *Turbulent Shear Flows II*, ed. Bradbury, Durst, Launder, Schmidt and Whitelaw,Springer-Verlag (1980).

38. Roache, P.J., *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico (1976).

39. Runchal, A. K., "CONDIF: A Modified Central-Difference Scheme for Convective Flows", *International Journal for Numerical Methods in Engineering* **24** pp. 1593-1608 (1987).

40. Seyed, S.A., Gosman, A.D., and Peric, M., "Assessment of Discretisation Schemes to Reduce Numerical Diffusion in the Calculation of Complex Flows", *AIAA-85-0441*, (1985).

41. Shah, M., "Private Communication", *Oxford University Computing Laboratory*, (1988).

42. Sharif, M.A.R. and Busnaina, A.A., "Assessment of Finite Difference Approximations for the Advection Terms in the Simulation of Practical Flow Problems", *Journal of Computational Physics* **74** pp. 143-176 (1988).

43. Shaw, G. and Sivaloganathan, S., "On the Smoothing Properties of the SIMPLE Pressure-Correction Algorithm", *International Journal for Numerical Methods in Fluids* **8** pp. 441-461 (1988).

44. Shyy, W., "A Study of Finite Difference Approximations to Steady-State, Convection-Dominated Flow Problems", *Journal of Computational Physics* **57** pp. 415-438 (1985).

45. Sivaloganathan, S. and Shaw, G.J., "An Efficient Non-linear Multigrid Procedure for the Incompressible Navier-Stokes Equations", pp. 221-233 in *Proceedings of the 5th International Conference on Numerical Methods in Laminar and Turbulent Flow, Part 1*, ed. Taylor, Habashi and Hafez, (1987).

46. Spalding, D.B., "A Novel Finite Difference Formulation for Differential Expressions Involving both First and Second Derivatives", *International Journal for Numerical Methods in Engineering* **4** pp. 551-559 (1972).

47. Thompson, J.F., "Three Dimensional Adaptive Grid Generation on a Composite Block Grid", in *Numerical Methods in Fluid Dynamics - in press*, ed. Morton and Baines,Clarendon Press, Oxford ().

48. Timin, T. and Esmail, M.N., "A Comparative Study of Central and Upwind Difference Schemes Using the Primitive Variables", *Internatinal Journal Methods in Fluid* **3** pp. 295-305 (1983).

49. Vanka, S.P., "Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables", *Journal of Computational Physics* **65** pp. 138-158 (1986).

50. Zedan, M. and Schneider, G.E., "Investigation of Simultaneous Variable Solution Procedure for Velocity and Pressure in Incompressible Fluid Flow Problems", *AIAA-83-1519*, (June 1983).

51. Zedan, M. and Schneider, G.E., "A Strongly Implicit Simultaneous Variable Solution Procedure for Velocity and Pressure in Fluid Flow Problems", *AIAA-83-1569*, (June 1983).

52. Zhu, J. and Leschziner, M.A., "A Local Oscillation-Damping Algorithm for Higher-Order Convection Scheme", *Submitted to International Journal for Computational Methods in Applied Mechanics and Engineering*, (1987).

Chapter 2

DISCRETISATION OF THE GOVERNING EQUATIONS

2.1. Introduction

In this chapter several important aspects of numerical approximations to incompressible flow are considered in detail, before proceeding to the main body of the text. It is instructive to consider, without loss of generality, the equation governing the transport of a scalar quantity, ϕ, through such a fluid, which can be written in Cartesian co-ordinates as

$$\frac{\partial \phi}{\partial t} + \frac{\partial (u_j \phi)}{\partial x_j} = \frac{\partial}{\partial x_j}(\Gamma \frac{\partial \phi}{\partial x_j}) + S_\phi \qquad (2.1)$$

$$\begin{bmatrix}\text{time rate}\\ \text{of change}\end{bmatrix} + \begin{bmatrix}\text{convective}\\ \text{term}\end{bmatrix} = \begin{bmatrix}\text{diffusion}\\ \text{term}\end{bmatrix} + \begin{bmatrix}\text{source}\\ \text{term}\end{bmatrix},$$

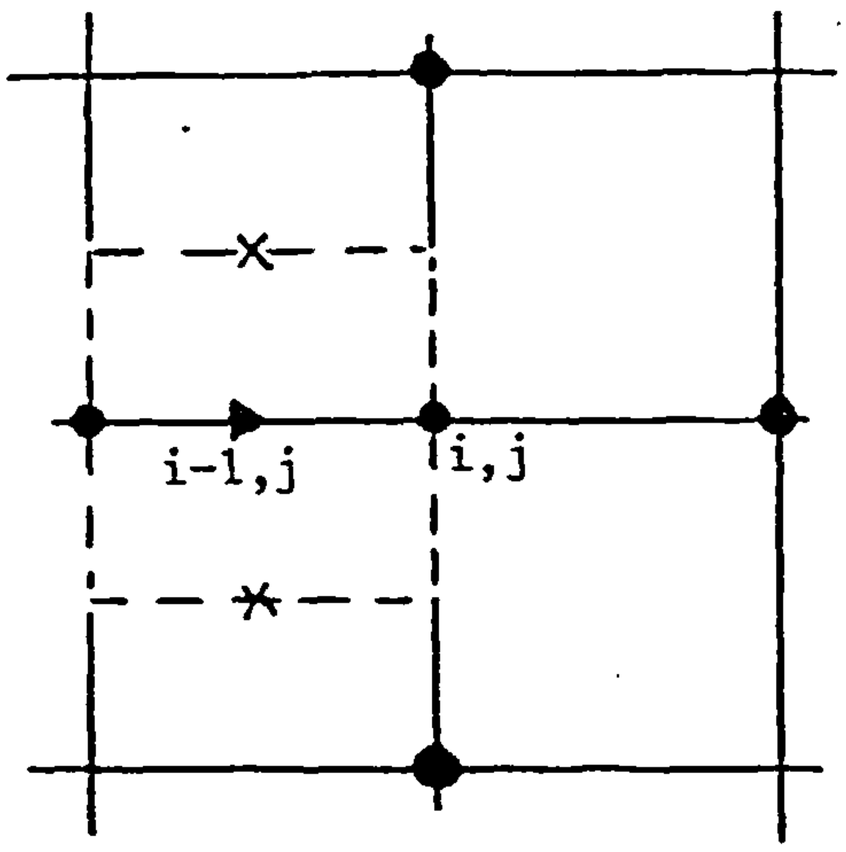
where u_i is the velocity component in the x_i direction, Γ is a constant scalar diffusion coefficient and S_ϕ represents a source term for ϕ. We shall now use this equation to highlight the fundamental ideas behind discretisation techniques before proceeding to the equations governing fluid flow, in Chapter 3.

In order to find a numerical solution to equation (2.1) we must construct a set of algebraic equations whose solution give a discrete representation of the continuum problem. In line with accepted practice, first introduced by Harlow and Welch[5], the solution domain is divided up into a series of contiguous finite control volumes. A set of algebraic equations is then constructed for each control volume based on a staggered grid arrangement - the scalar quantity is calculated at a point in the centre of each control volume, whereas velocity components are calculated at the interfaces (see Figure 2.1).

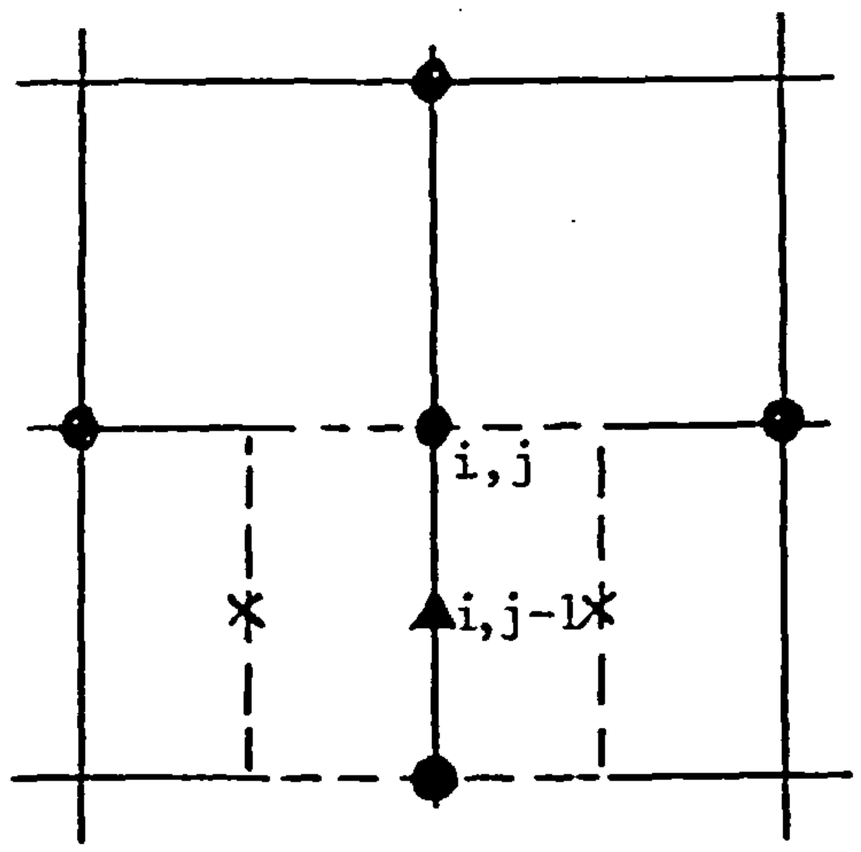
For steady-state flows the time derivative in equation (2.1) disappears and if we consider the absence of any source terms we get

$$\frac{\partial u_j \phi}{\partial x_j} = \frac{\partial}{\partial x_j}(\Gamma \frac{\partial \phi}{\partial x_j}) \qquad (2.2)$$

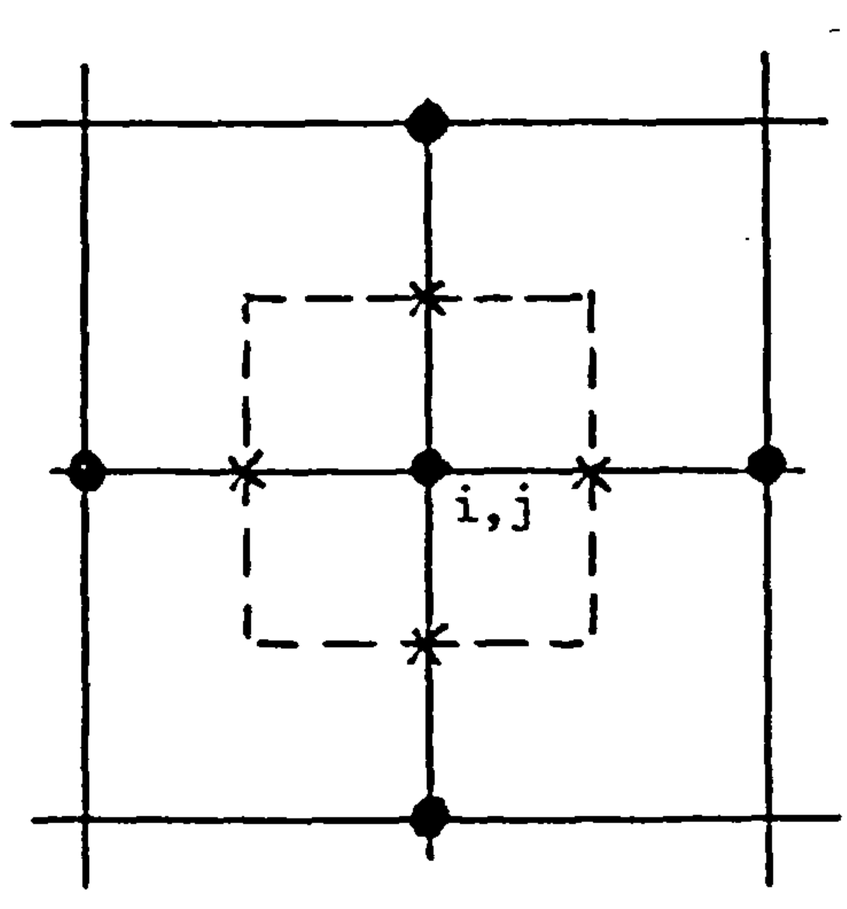
which when integrated over a control volume gives



u - momentum



v - momentum



scalar

Figure 2.1: Staggered grid arrangement for velocity and scalar control volumes on a two-dimensional grid.

$$\int \frac{\partial u_j \phi}{\partial x_j}\, dx_1 dx_2 = \int \frac{\partial}{\partial x_j} \Gamma \frac{\partial \phi}{\partial x_j}\, dx_1 dx_2. \qquad (2.3)$$

Writing this in two-dimensional component form and expanding one gets

$$\int \frac{\partial}{\partial x} u\phi \, dxdy + \int \frac{\partial}{\partial y} v\phi \, dxdy = \int \frac{\partial}{\partial x} \Gamma \frac{\partial \phi}{\partial x} \, dxdy + \int \frac{\partial}{\partial y} \Gamma \frac{\partial \phi}{\partial y} \qquad (2.4)$$

which gives, (assuming a square control volume),

$$\frac{1}{h}((u\phi)_{i+½j} - (u\phi)_{i-½j} + (v\phi)_{ij+½} - (v\phi)_{ij-½}) = \frac{1}{h}[(\Gamma \frac{\partial \phi}{\partial x})_{i+½j} - (\Gamma \frac{\partial \phi}{\partial x})_{i-½j} + (\Gamma \frac{\partial \phi}{\partial y})_{ij+½} (\Gamma \frac{\partial \phi}{\partial y})_{ij-½}]. \quad (2.5)$$

This equation represents the conservation of the scalar ϕ over this control volume, each term representing a particular flux.

The next step is to find a way of approximating the face values of ϕ namely $\phi_{i+½j}$, $\phi_{i-½j}$, $\phi_{ij+½}$, and $\phi_{ij-½}$. This is done by constructing an interpolating polynomial in terms of neighbouring nodes. Several schemes for doing this have been proposed, some of which are outlined later. All concentrate on the correct modelling of the convective term $\frac{\partial u_i \phi_j}{\partial x_j}$, which is by far the most troublesome. Ideally any numerical model for convection should comply with the prerequisite set of desirable properties outlined in section 2.2.

Discretisation of the diffusion terms is by comparison relatively simple; central differencing is invariably used, as it is throughout this thesis.

2.2. Fundamental Principles

One reason for the large number of different schemes in existence is the apparent lack of rigorous testing techniques. Experimental results are difficult to obtain for many of the flows considered, the primary reason for using numerical methods in the first instance being the prohibitive cost of experimentation. Several test problems have been reported, but direct comparisons are rarely made. The test problems employed are often simple (one-dimensional, linear etc.) and as such the results do not necessarily correlate with or reflect the characteristics of many complex flows of interest.

Several concepts have been used to evaluate the schemes and these are discussed below.

2.2.1. Accuracy of Discretisation

There are two measures used for assessing accuracy of discrete approximations. They are 'order of truncation error' and 'order of interpolation error in the approximating polynomial' - applied to finite difference and finite volume schemes, respectively.

The truncation error is defined as the difference between the differential operator and the Taylor Series expansion of the difference operator; that is

$$T_{ij} = \bar{L}_{ij} - L_{ij}, \tag{2.6}$$

where $\bar{L}_{ij}$ and L_{ij} are the differential operator and difference operator, respectively. For example, consider the well known central differencing formula

$$\overline{(\frac{\partial \phi}{\partial x})}_{ij} = \frac{\phi_{i+1j} - \phi_{i-1j}}{2h}. \tag{2.7}$$

Expanding this in terms of a Taylor Series gives

$$\phi_{i\pm1j} = \phi_{ij} \pm h(\frac{\partial \phi}{\partial x})_{ij} + \frac{h^2}{2}(\frac{\partial^2 \phi}{\partial x^2})_{ij} \pm \frac{h^3}{6}(\frac{\partial^3 \phi}{\partial x^3})_{ij} + O(h^4). \tag{2.8}$$

So that

$$\overline{(\frac{\partial \phi}{\partial x})}_{ij} = \frac{1}{2h}\left[2h(\frac{\partial \phi}{\partial x})_{ij} + \frac{h^3}{3}(\frac{\partial^3 \phi}{\partial x^3})_{ij} + O(h^5)\right], \tag{2.9}$$

and therefore

$$T_{ij} = \overline{(\frac{\partial \phi}{\partial x})}_{ij} - (\frac{\partial \phi}{\partial x})_{ij} = \frac{h^2}{6}(\frac{\partial^3 \phi}{\partial x^3})_{ij} + O(h^4). \tag{2.10}$$

Accordingly central differencing is said to be of order two or second order accurate.

With a control volume formulation, the order is defined as the order of the error of the interpolation used. Thus linear interpolation (which corresponds to central differencing) is of order two: this is the same as the order of the truncation error.

However, these two definitions do not give the same values for higher order schemes.

It is doubtful however, whether the concept of truncation error is relevant to highly complex fluid flows. Shyy[15, 17] has shown that it is only meaningful for those Fourier components of the real solution having a wave number$\mathcal{O}(2\pi/h)$. For higher wave numbers, $\mathcal{O}(\frac{\pi}{h})$, the error is independent of the mesh size and the order of accuracy is irrelevant. If rapid changes of ϕ occur, they cannot be resolved over one mesh spacing unless h is prohibitively small. A decrease in h will only cut down the occurrence of error regions, but can never resolve arbitrarily steep gradients that often occur. So truncation error may not, in general, give a good indication of the accuracy of an approximation - see, for example, Gaskell and Lau[2].

2.2.2. Convective Stability

The stability concept used for time dependent flows can be extended to steady-state computations merely by considering the iteration number to represent time[7].

Consider the nodal value ϕ_i. Any disturbance to ϕ_i from outside influences must reduce or enhance C_i (the flux into the control volume centred on i) in accordance with whether ϕ_i increases or decreases in order to ensure convective stability. In short, the scheme must have negative feedback. This concept can be stated mathematically as follows:

Define the 'feedback function' of the convective influx to be $\frac{\partial C_i}{\partial \phi_i}$, then

$$\frac{\partial C_i}{\partial \phi_i} \begin{cases} < 0 & \text{stable sensitivity} \\ = 0 & \text{neutral sensitivity} \\ > 0 & \text{unstable sensitivity} \end{cases} . \tag{2.11}$$

Leonard[6] has shown that any numerical approximation to convection not possessing an element of upwind bias cannot have convective stability and vice versa. From a physical viewpoint we can see that convection is associated with the transport of fluid from upstream to downstream, therefore any numerical approximation should reflect this property.

As well as convective stability there is the analogous property of diffusive stability; that is the sensitivity of D_{IF}, the diffusive flux, to changes in ϕ_i . When the diffusive term is approximated by central differencing, this is always negative. Sometimes this is sufficient to counter-balance a positive convective sensitivity, but for flows where the cell Reynolds number (uh/ν also known as the Peclet number) is high (convection dominated) it is inadequate.

2.2.3. Conservative Form

A discretisation scheme is said to be conservative if there are no effective source terms in the algebraic analogue, that do not appear in the governing partial differential equation. If this is the case, then the algebraic equation mirrors the conservative property of the differential equation exactly. This can be stated mathematically as follows.

If $\overline{\frac{\partial\phi}{\partial x}}$ is the approximation to $\frac{\partial\phi}{\partial x}$ then the discretisation scheme is conservative if it is of the form;

$$\overline{\frac{\partial\phi}{\partial x}} = \frac{1}{\delta x}[H(\phi_{i+l},.....,\phi_{i-l+1}]-H(\phi_{i+l-1},.....,\phi_{i-1})], \qquad (2.12)$$

where H is a function of 2l arguments which must for consistency satisfy

$$H(\phi_i,\phi_i,.....,\phi_i) = \phi_i.$$

Or put more simply, conservation is obeyed if the computed flux through the right hand side of the control volume centred at i,j is equivalent to the flux through the left hand side of the control volume centred at i+1,j. The concept of conservation is not as important for when sources are present as it is for cases in which they are absent, because erroneous sources will be introduced by source term approximations anyway.

2.2.4. Diagonal Dominance

All the discretisation schemes outlined section 2.3 produce systems of algebraic equations that have to be solved by iterative techniques. Therefore, the form of the coefficient matrix generated by these techniques can be important. It can be seen that

solutions, ϕ, to the scalar transport equation in the absence of source terms must lie within the bounds $\min(\phi_B)$ and $\max(\phi_B)$ where ϕ_B is the boundary function. Any discretisation scheme must reflect this. A sufficient condition for satisfaction of this property is that the solution matrix is diagonally dominant, that is, $|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \forall i$. The property of diagonal dominance also ensures that the coefficient matrix is numerically stable. Attempts to satisfy this criterion have dominated much of the recent research into developing a new generation of high-order bounded schemes[13, 14, 19] for convective transport.

2.2.5. Boundedness

In terms of the above definition, many of the higher-order schemes presently available are unbounded, yet they are found often to generate perfectly acceptable results, if the conditions are favourable. Gaskell and Lau state[4] "It is arguable therefore that strict adherence to diagonal dominance and its satisfaction as a prerequisite for bounded solutions could be misleading since it represents, in some cases, only a sufficient condition for guaranteeing boundedness." Consequently they have suggested a more physically relevant definition of boundedness, by using the idea of interpolative boundedness. Consider a control volume centred at i-1 and define a normalised face value, $\hat{\phi}_k$ as

$$\hat{\phi}_k = \frac{(\phi_k - \phi_{i-2})}{(\phi_i - \phi_{i-2})}, \tag{2.13}$$

then for interpolative boundedness $\hat{\phi}_{i-\frac{1}{2}}$ must lie within the bounds of its neighbouring values at the nodes i & i-1 (given that there is no source present). So it is required that $\hat{\phi}_{i-\frac{1}{2}} \in (\hat{\phi}_{i-1}, 1]$ when $\hat{\phi}_{i-1} \in (-\infty, 1]$ and $\hat{\phi}_{i-\frac{1}{2}} \in [1, \hat{\phi}_{i-1})$ when $\hat{\phi}_{i-1} \in [1, \infty)$.

Interpolative boundedness, as defined here, is a necessary but not sufficient condition, for computed boundedness. Gaskell and Lau go further and define a Convective Boundedness Criterion (CBC) for implicit steady state flow conditions, that ensures computed boundedness.

Criterion Define a continuous increasing function or a union of piecewise continuous increasing functions, f relating the modelled normalised face value, $\hat{\phi}_{i-½}$ to the normalised upstream nodal value, $\hat{\phi}_{i-1}$, that is $\hat{\phi}_{i-½} = f(\hat{\phi}_{i-1})$, then a finite difference approximation to $\phi_{i-½}$ is bounded if:-

i) for $\hat{\phi}_{i-1}\epsilon[0,1]$, f is bounded below by the function $\hat{\phi}_{i-½}=\hat{\phi}_{i-1}$ and above by unity and passes through the points (0,0) and (1,1).

ii) for $\hat{\phi}_{i-½}\notin[0,1]$ f is equal to $\hat{\phi}_{i-1}$.

This is shown graphically in Figure 2.2. It differs from the property of diagonal dominance in that it is based on physical reasoning rather than mathematical abstraction. It represents a radical shift of emphasis and forms the basis of the Curvature Compensated Convective Transport (CCCT) approximation outlined in the next section.

2.3. Methods of Discretising the Convective Term

It is worthwhile reviewing the various discretisation schemes in prominent use for the prediction of recirculating flow. These are examined and discussed in terms of the concepts introduced above, with reference to Table 2.1 and Figure 2.3 The former lists some well known discretisation schemes and their associated properties while the latter shows their corresponding normalised profiles in relation to the CBC.

2.3.1. First-order interpolation

As a first attempt it would seem logical to use linear interpolation, let

$$\phi(x) = a + bx. \tag{2.14}$$

This gives,

$$\phi_{i+½j} \simeq \frac{\phi_{i+1j}+\phi_{ij}}{2}, \tag{2.15}$$

and

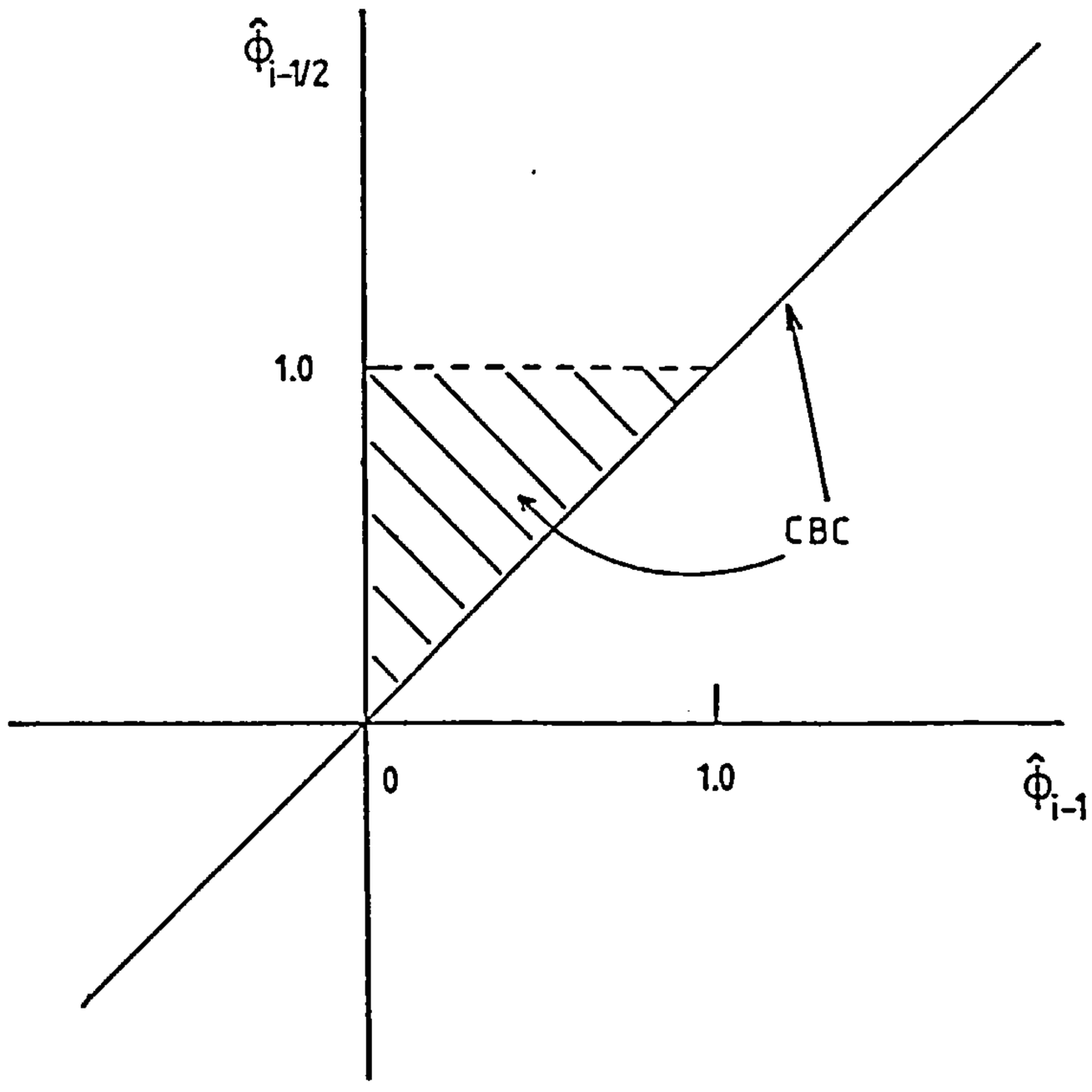


Figure 2.2: Diagrammatic representation of the convection boundedness criterion. The line $\hat{\phi}_{i-½}=\hat{\phi}_{i-1}$ and the shaded area indicate the region over which the criterion is valid.

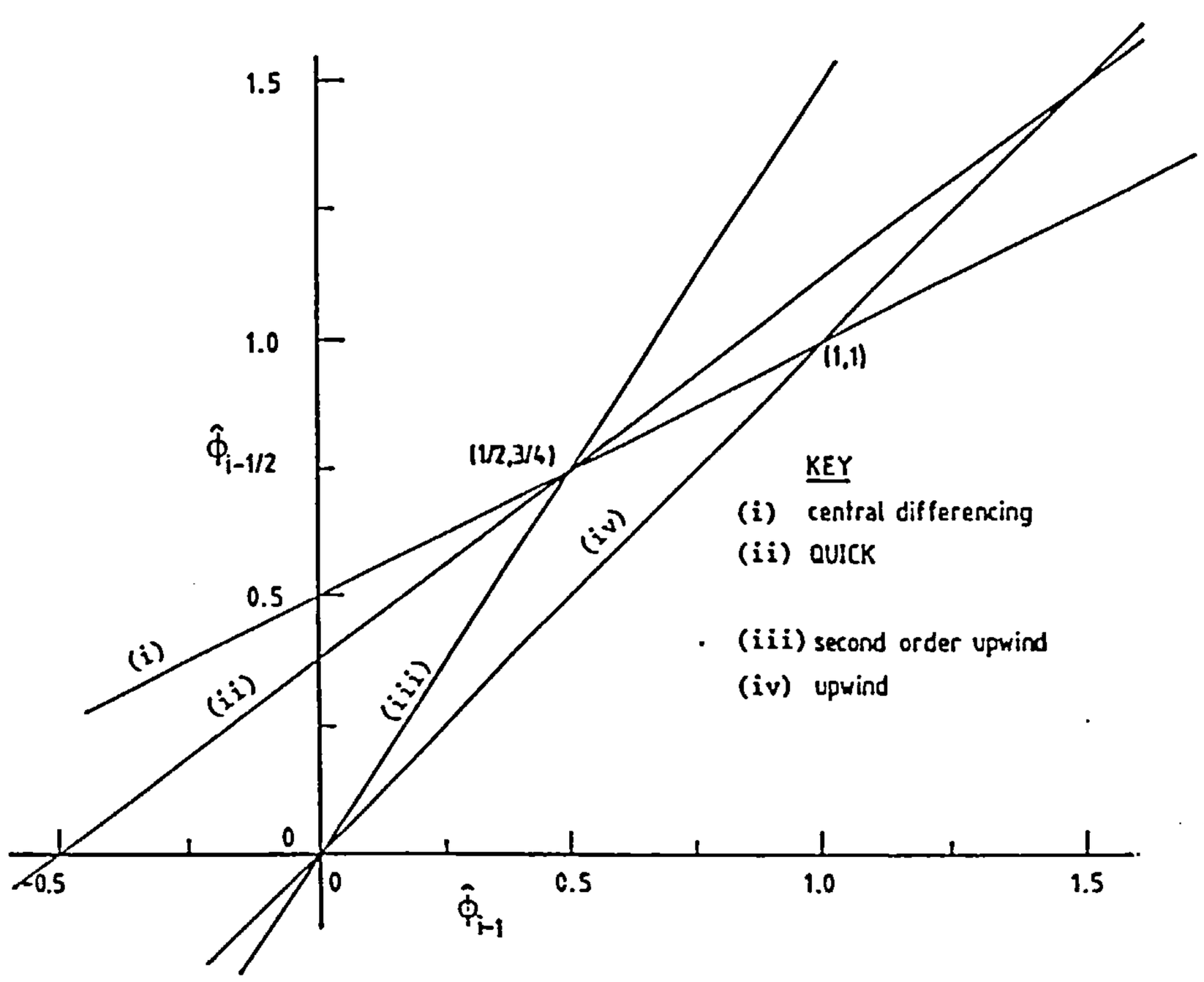


Figure 2.3: Normalised values of $\hat{\phi}_{i-½}$ for various well known approximations.

Discretization scheme	Finite difference expression for $\phi_{i-1/2}$ when $u_{i-1/2}=u_0>0$	Leading truncation error term	Convective stability $\partial C_{IF}/\partial\phi_{i-1}$	Boundedness Computed	Boundedness Interpolative	Critical Peclet number	α
QUICK	$\frac{1}{8}(3\phi_i+6\phi_{i-1}-\phi_{i-2})$	$\frac{u_0\Delta x^3}{16}\phi'''_{i-1/2}$	$-\frac{3u_0}{8\Delta x}$(stable)	No	No	$\frac{8}{3}$	0
Second-order upwind	$\frac{1}{2}(3\phi_{i-1}-\phi_{i-2})$	$\frac{3u_0\Delta x^2}{8}\phi''_{i-1/2}$	$-\frac{3u_0}{2\Delta x}$(stable)	No	No	∞	$\frac{3}{8}$
Central differencing	$\frac{1}{2}(\phi_i+\phi_{i-1})$	$\frac{u_0\Delta x^2}{4}\phi''_{i-1/2}$	0 (neutral)	No	Yes	2	$-\frac{1}{8}$
Upwind	ϕ_{i-1}	$\frac{u_0\Delta x}{2}\phi'_{i-1/2}$	$-\frac{u_0}{\Delta x}$(stable)	Yes	Yes	∞	—

Table 2.1: Some well known descretisation schemes and their associated properties.

$$\phi_{i-\frac{1}{2}j} \simeq \frac{\phi_{ij}+\phi_{i-1j}}{2},$$

similar expressions being obtained for $\phi_{ij-\frac{1}{2}}$ and $\phi_{ij+\frac{1}{2}}$.

For the case of constant velocity U_0 the left hand side of (2.7) reduces to

$$U_0\left[\frac{\phi_{i+\frac{1}{2}j}-\phi_{i-\frac{1}{2}j}}{h}\right], \tag{2.16}$$

and using linear interpolation this becomes

$$\frac{\phi_{i+1j}-\phi_{i-1j}}{2h}, \tag{2.17}$$

and so in finite difference terms we have 'central' differencing.

The central differencing scheme has been used extensively in engineering situations for the numerical solution of partial differential equations with even order derivatives. It gives high accuracy (truncation error order two) and has a minimum of computational complexity. However, its use for the odd-ordered derivatives found in convection problems leads to unrealistic solutions or non-convergence of the iterative scheme. Several authors[18,9] have found that central differencing leads to divergence, or that the solution is seen to contain unphysical oscillations, popularly known as 'wiggles'. The non-convergence can be attributed to the neutral convective stability of central differencing; an erroneous accumulation of the scalar ϕ is not counteracted by the scheme. Roache[12] showed that central differencing gives physically unrealistic solutions when the Peclet number is greater than 2 (this is referred to as the critical Peclet number). Central differencing is interpolatively, but not computationally bounded.

2.3.2. Zeroth-order Interpolation - Upwind Differencing

From a physical view point, advection is associated with the transport of a scalar from upstream to downstream and any numerical approximation should reflect this and so must possess an element of upwind bias. So faced with the problems of central differencing, zeroth-order interpolation was proposed as an alternative. Let

$$\phi(x) = \text{constant}, \tag{2.18}$$

giving

$$\phi_{i+\frac{1}{2}j} \simeq \phi_{ij} \quad \text{if } u>0, \tag{2.19}$$

$$\phi_{i+\frac{1}{2}j} \simeq \phi_{i+1j} \quad \text{if } u<0,$$

$$\phi_{i-\frac{1}{2}j} \simeq \phi_{i-1j} \quad \text{if } u>0,$$

$$\phi_{i-\frac{1}{2}j} \simeq \phi_{ij} \quad \text{if } u<0.$$

Once again consider the constant velocity case, with $U_0 > 0$, then

$$U_0\left[\frac{\phi_{i+\frac{1}{2}j}-\phi_{i-\frac{1}{2}j}}{h}\right] \simeq U_0\left[\frac{\phi_{ij}-\phi_{i-1j}}{h}\right], \tag{2.20}$$

which corresponds to a backward finite difference formula. The 'critical' Peclet number in this case is infinity and the convective stability is less than zero. It is both computationally and interpolatively bounded and is conservative. So theoretically solutions should always be obtainable and convergence should be guaranteed. This has been borne out in practise by many authors. However, this is only a first order technique and investigation of the truncation error reveals some interesting reasons for this stability or 'robustness'. Consider the one dimensional linear equation

$$u\frac{\partial\phi}{\partial x}=\Gamma\frac{\partial^2\phi}{\partial x^2}, \qquad u>0. \tag{2.21}$$

We shall use central differencing for diffusion and upwind for convection. Define the truncation error as

$$T_i=u(\overline{\frac{\partial\phi}{\partial x}})_i - \Gamma(\overline{\frac{\partial^2\phi}{\partial x^2}})_i \ , \tag{2.22}$$

and substitute in the Taylor series for the two derivatives, to give

$$T_i=u\frac{\partial\phi}{\partial x}+\frac{uh}{2}\frac{\partial^2\phi}{\partial x^2}-\Gamma\frac{\partial^2\phi}{\partial x^2}+\mathcal{O}(h^2) \tag{2.23}$$

$$=u\frac{\partial\phi}{\partial x}-\Gamma\frac{\partial^2\phi}{\partial x^2}(1+\frac{uh}{2\Gamma})+O(h^2).$$

It can be seen that the truncation error of the upwind scheme corresponds to an 'artificial' or 'numerical' diffusion, Indeed this term becomes larger than the natural diffusion when $\frac{uh}{2\Gamma}>1$ or when Pe>2, that is just in the regions where central is unstable. It is this numerical diffusion that accounts for the stability of the upwind scheme. So, it is clear that upwind's stability advantage over central differencing is gained at the expense of accuracy. As stated by Roache[12], to call this a solution to the instability is rather fictitious. It merely represents the introduction of a damping factor. Gaskell and Lau[2] have observed that this numerical diffusion completely obscures the features of some fluid flows, proving to be particularly poor for turbulent flow situations[3].

2.3.3. The hybrid scheme

In 1972 Spalding proposed the hybrid scheme[16] as a compromise. This is a combination of the upwind and central differencing schemes described above, with one or the other being used on the basis of a specified criterion; namely, if Pe<2 then central differencing is used, otherwise upwind differencing is used and diffusion is neglected.

Clearly, this scheme reduces to upwind for high Reynolds number flows. However, it is important to note that despite its inherent inaccuracy it has been used extensively over the past 16 years in both academic and commercial spheres. In fact it forms the basis of several popular software packages. The essential feature of the hybrid scheme (not shown in Table 2.1 since it is either equivalent to central or upwind differencing) is that central differencing is accurate and stable when the Peclet number is below 2 (and so is used then) otherwise upwind which is stable for all Peclet numbers is used . So it is accurate when Pe<2, but for all other values it has the same deficiencies as upwind. It's popularity is based on its ability to generate bounded solutions in many different flow regimes. On the other hand the above analysis shows that this advantage is based on somewhat dubious grounds.

2.3.4. The Skew Upwind Differencing Scheme (SUDS)

Raithby proposed this scheme in 1976[10], following a wide-ranging investigation by him of the main deficiencies of the upwind approximation[11]. His goal was to establish a better approximation to convective transport in regions where the grid line and velocity direction are not closely aligned; a situation that upwinding is known to deal with rather poorly. He assumes a local profile of the form

$$\phi(x,y) = C_1 + C_2(y\frac{u}{V} - x\frac{v}{V}), \tag{2.24}$$

where (u,v) is the prevalent velocity at the face where the value of ϕ is required and $V=\sqrt{u^2+v^2}$, C_1 and C_2 being obtained from values located upstream from the chosen face. The scheme itself is rather unwieldy and the reader is referred to the author's original text for further details of its implementation; associated results can also be found in Raithby's paper[10].

Later Raithby proposed a further refinement called the Skew Upstream Weighted Differencing Scheme (SUWDS). The scheme has a third term in the interpolating polynomial

$$C_3 \exp(\frac{\rho u x}{\Gamma} + \frac{\rho v y}{\Gamma}). \tag{2.25}$$

This is intended to improve approximations when Pe≈1, that is when both convection and diffusion play an important part in determining the flow field. However, the implementation of such schemes can be very complicated and they only go part of the way towards curing the ills of upwind differencing, since they are not fully bounded. This has lead to the development of even higher order approximations to convective transport, the most significant of which are reviewed below.

2.3.5. Second Order Upwind Differencing

This scheme was proposed by Atias, Wolfstein and Israel[1] in 1977. In order to approximate the flux $\phi_{i-½j}$ they used the two upwind points located at ϕ_{i-1j} and ϕ_{i-2j}. In which case

$$\phi_{i-\frac{1}{2}j} = \frac{3\phi_{i-1j}-\phi_{i-2j}}{2} \quad \text{if } u>0. \tag{2.26}$$

For constant velocity, $U_0>0$, this gives

$$(u\phi)_{i+\frac{1}{2}j}-(u\phi_{i-\frac{1}{2}j}) = \frac{U_0}{2}(3\phi_{ij}-4\phi_{i-1j}+\phi_{i-2j}). \tag{2.27}$$

This scheme is more accurate (interpolative error equal to two) than upwinding, and this improvement has been seen practically. But, it is prone to oscillatory solutions and is neither computationally nor interpolatively bounded. Even so this scheme is receiving renewed attention by turbulence modellers, several of whom have been reported as acknowledging it to be the best approach in turbulent flow simulations[8].

2.3.6. Quadratic Upwind Interpolation

In 1979 Leonard[6] proposed the Quadratic Upwind Interpolation for Convective Kinematics (QUICK) scheme. This employs an upwind biased quadratic profile to approximate face values.

Assume a quadratic profile, $\phi(x)=a+bx+cx^2$ through ϕ_{ij}, ϕ_{i-1j} and ϕ_{i-2j}. This gives

$$\phi_{i-\frac{1}{2}j}=\frac{3}{8}\phi_{ij}+\frac{3}{4}\phi_{i-1j}-\frac{1}{8}\phi_{i-2j}. \tag{2.28}$$

With constant velocity, $U_0>0$,

$$(u\phi)_{i+\frac{1}{2}j}-(u\phi_{i-\frac{1}{2}j}) = \frac{U_0}{8}(3\phi_{i+1j}+3\phi_{ij}-7\phi_{i-1j}+\phi_{i-2j}). \tag{2.29}$$

A two-dimensional version of the QUICK scheme can be derived using a polynomial in both x and y, as follows

$$\phi(x,y)=a+bx+cy+dx^2+fxy+ey^2. \tag{2.30}$$

So

$$\phi_{i-\frac{1}{2}j}=\int_{-\frac{1}{2}}^{\frac{1}{2}}\phi(-\tfrac{1}{2},y)\,dy \tag{2.31}$$

$$= \int_{-\frac{1}{2}}^{\frac{1}{2}} \left[a - \frac{b}{2} + cy + \frac{d}{4} - \frac{ex}{4} + fy^2 \right] dy$$

$$= \frac{3}{8}\phi_{ij} + \frac{3}{4}\phi_{i-1j} - \frac{1}{8}\phi_{i-2j} + \frac{1}{24}(\phi_{i-1j-1} - 2\phi_{i-1j} + \phi_{i-1j+1}).$$

It can also be extended to three dimensions.

Most authors wishing to use a higher order technique have chosen this one. It has an interpolative error of order three and is conservative. Also, it has been shown[4] to accurately predict complicated fluid phenomena that are missed by the hybrid technique. It has no formal numerical diffusion, but it is not bounded. It is found to produce oscillatory solutions in regions of sharp changes in gradient of a dependent variable, but in the main it is very accurate and computationally efficient.

2.3.7. Curvature Compensated Convective Transport - CCCT

It can be seen from the above discussion that the answer to the question "Which discretisation shall we use?" lies in two places. On the one hand we have the hybrid scheme which is bounded and stable but often highly diffusive. On the other hand we have QUICK which is accurate and non-diffusive, but which is unbounded.

Gaskell and Lau[4] have outlined a solution to this dichotomy. They set about devising a scheme that was both high order and non-diffusive, but also bounded. They have suggested a scheme that is third order, where the definition of order is meaningful, but that also deals with areas of steep gradients(where QUICK fails) in a physically realistic manner. This scheme satisfies the Convection Boundedness Criterion outlined above.

Although its derivation as rather complex, the algorithm itself is incredibly simple and easy to program. For u>0 we have that

$$\phi_{i-\frac{1}{2}j} = \left[\frac{3}{4} + 2\alpha\right]\phi_{ij} + \left[\frac{3}{8} - \alpha\right]\phi_{i-1j} - \left[\frac{1}{8} + \alpha\right]\phi_{i-2j} \qquad (2.32)$$

where

$$\alpha = \left[\frac{\hat{\phi}_{i-\frac{1}{2}j} - \frac{3}{8}\left[2\hat{\phi}_{i-1j}+1\right]}{2\hat{\phi}_{i-1j}-1} \right]$$

$$\hat{\phi}_{i-\frac{1}{2}j} = \begin{cases} \hat{\phi}_{i-1j} & \text{and } \alpha \in (-1/8,3/8) \text{ if } \hat{\phi}_{i-1j} \notin [0,1] \\ 3\hat{\phi}_{i-1j} & \text{and } \alpha \in (0,3/8] \text{ if } \hat{\phi}_{i-1j} \in [0,1/6) \\ 1 & \text{and } \alpha \in [-1/8,0) \text{ if } \hat{\phi}_{i-1j} \in (5/6,1] \\ \frac{3}{8}(2\hat{\phi}_{i-1j}+1) & \text{and } \alpha=0 \text{ if } \hat{\phi}_{i-1j} \in [1/6,5/6] \end{cases}$$

This is an appropriate point to remind ourselves of the object of the present work; namely to develop a robust , flexible and accurate multigrid solution strategy for predicting fluid flows. Clearly the discretisation approach adopted represents a crucial part of any such approach, and thus warrants careful consideration. CCCT reflects all of the desirable attributes discussed in section 2.2 and in view of this was considered to represent the most appropriate choice.

Also CCCT represents a generic approximation to convection in that it can be used to generate several of the other schemes described above, by fixing the value of α - see Table 2.1. Last, but not least, the originators of CCCT have satisfactorily applied it to a wide range of turbulent flow problems which is an important factor regarding the extension of the present methodology to the treatment of such flows.

Note that since the problems considered in subsequent chapters do not exhibit a lack of boundedness[2], α can be set equal to zero to yield maximum accuracy.

References

1. Atias, M., Wolfshtein, M., and Israeli, M., "Efficiency of Navier-Stokes Solvers", *AIAA Journal* **15**(2) pp. 263-266 (1977).

2. Gaskell, P.H. and Lau, A.K.C., "An Efficient Solution Strategy for Use with Higher Order Discretisation Schemes", *Report No. T40, Department of Mechanical Engineering, University of Leeds*, (1986).

3. Gaskell, P.H. and Lau, A.K.C., "An Assessment of Direct Stress Modelling for Elliptic Turbulent Flows with the Aid of a Non-Diffusive Boundedness

Preserving Discretisation Scheme'', pp. 351-362 in *Proceedings of The 5th International Conference on Numerical Methods in Laminar and Turbulent Flow, Part 1*, ed. Taylor, Habashi and Hafez, (1987).

4. Gaskell, P.H. and Lau, A.K.C., ''Curvature Compensated Convective Transport : SMART, a New Boundedness Preserving Transport Algorithm'', *International Journal for Numerical Methods in Fluids* **8** pp. 617-641 (1988).

5. Harlow, F.H. and Welch, J.E., ''Numerical Calculation if Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface'', *The Physics of Fluids* 8(12) pp. 2183-2189 (1965).

6. Leonard, B.P., ''A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation'', *Computational Methods in Applied Mechanics and Engineering* **19** pp. 59-98 (1979).

7. Leonard, B.P., ''The QUICK Algorithm : A Uniformly Third Order Method for Highly Convective Flows'', pp. 159 in *Computer Methods in Fluids*, ed. Morgan, Taylor & Brebbia,Pentech Press (1980).

8. Nallasamy, M., ''Turbulence Models and their Application to the Prediction of Industrial Flows: A Review'', *Computers and Fluids* **15**(2) pp. 151-194 (1987).

9. Patel, M.K., Markatos, N.C., and Cross, M., ''A Critical Evaluation of Seven Discretisation Schemes for Convection-Diffusion Equation'', *International Journal for Numerical Methods in Fluids* **5** pp. 225-244 (1985).

10. Raithby, G.D., ''Skew Upstream Differencing Schemes for Problems Involving Fluid Flow'', *Computer Methods in Applied Mechanics and Engineering* **9** pp. 153-164 (1976).

11. Raithby, G.D., ''A Critical Evaluation of Upstream Differencing applied to Problems involving Fluid Flow'', *Computer Methods in Applied Mechanics and Engineering* **9** pp. 75-103 (1976).

12. Roache, P.J., *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico (1976).

13. Runchal, A. K., "CONDIF: A Modified Central-Difference Scheme for Convective Flows", *International Journal for Numerical Methods in Engineering* **24** pp. 1593-1608 (1987).

14. Seyed, S.A., Gosman, A.D., and Peric, M., "Assessment of Discretisation Schemes to Reduce Numerical Diffusion in the Calculation of Complex Flows", *AIAA-85-0441*, (1985).

15. Shyy, W., "A Study of Finite Difference Approximations to Steady-State, Convection-Dominated Flow Problems", *Journal of Computational Physics* **57** pp. 415-438 (1985).

16. Spalding, D.B., "A Novel Finite Difference Formulation for Differential Expressions Involving both First and Second Derivatives", *International Journal for Numerical Methods in Engineering* **4** pp. 551-559 (1972).

17. Stubley, G.D., Raithby, G.D., and Strong, A.B., "Proposal for a New Discrete Method Based on an Assessment of Discretisation Errors", *Numerical Heat Transfer* **3** pp. 411-428 1980, ().

18. Timin, T. and Esmail, M.N., "A Comparative Study of Central and Upwind Difference Schemes Using the Primitive Variables", *Internatinal Journal Methods in Fluid* **3** pp. 295-305 (1983).

19. Zhu, J. and Leschziner, M.A., "A Local Oscillation-Damping Algorithm for Higher-Order Convection Scheme", *Submitted to International Journal for Computational Methods in Applied Mechanics and Engineering*, (1987).

Chapter 3

CHOOSING A SMOOTHING TECHNIQUE

3.1. Introduction

Having chosen a suitable discretisation procedure, bearing in mind the desirable attributes discussed in Chapter 2, a system of algebraic equations can be constructed to represent the continuum problem over some domain of interest.

This thesis is concerned specifically with the solution of the equations of motion which govern steady laminar flow of a viscous fluid, which when written in terms of dimensionless coordinates and the Reynolds number, Re, become[1]

$$\frac{\partial u_\alpha}{\partial x_\alpha} = 0, \tag{3.1a}$$

$$\frac{\partial}{\partial x_\beta}(u_\alpha u_\beta) = -\frac{1}{\rho}\frac{\partial p}{\partial x_\alpha} + \frac{\partial}{\partial x_\beta}\left[\frac{1}{Re}\frac{\partial u_\alpha}{\partial x_\beta}\right] \quad \alpha = 1,2,3 \tag{3.1b}$$

where p is the pressure, and u_α and x_α are the velocity and distance in the α coordinate direction respectively. Henceforth we shall dispense with the idea of a general transport equation.

Considering two-dimensional flow and integrating equations (3.1a and b) over a control volume centred at the node (i,j) we get

$$\frac{(u^2)_{i+\frac{1}{2}j} - (u^2)_{i-\frac{1}{2}j} + (vu)_{ij+\frac{1}{2}} + (vu)_{ij-\frac{1}{2}}}{h} = -\frac{1}{\rho}\frac{p_{i+1j}-p_{ij}}{h} \tag{3.2a}$$

$$+ \frac{1}{Re\,h^2}(u_{i+1j} + u_{i-1j} + u_{ij+1} + u_{ij-1} - 4u_{ij}),$$

$$\frac{(uv)_{i+\frac{1}{2}j} - (uv)_{i-\frac{1}{2}j} + (v^2)_{ij+\frac{1}{2}} + (v^2)_{ij-\frac{1}{2}}}{h} = -\frac{1}{\rho}\frac{p_{ij+1}-p_{ij}}{h} \tag{3.2b}$$

$$+ \frac{1}{Re\,h^2}(v_{i+1j} + v_{i-1j} + v_{ij+1} + v_{ij-1} - 4v_{ij}),$$

$$\frac{u_{ij} - u_{i-1j} + v_{ij} - v_{ij-1}}{h} = 0, \tag{3.2c}$$

where h, the mesh spacing, is taken to be the same in each coordinate direction (as is the case throughout this thesis). The control volumes corresponding to each of these

equations are shown in Figure 2.1 of the previous chapter. The velocities on the right hand side of equations (3.2a and b) are at the points corresponding to finite difference nodes and need no further approximation. The fluxes on the left hand side, however, are required at points between nodes and must be approximated. Each term is split into two parts - one involving the advecting velocity and the other the advected velocity. The latter is the same as the scalar variable ϕ used in Chapter 2. The advecting velocity is approximated by linear interpolation and the advected velocity by whatever scheme has been chosen. This gives,

$$F_{i+\frac{1}{2}j}u_{i+\frac{1}{2}j} - F_{i-\frac{1}{2}j}u_{i-\frac{1}{2}j} + F_{ij+\frac{1}{2}}u_{ij+\frac{1}{2}} - F_{ij-\frac{1}{2}}u_{ij-\frac{1}{2}} = -\frac{1}{\rho}\frac{p_{i+1j}-p_{ij}}{h} \tag{3.3}$$

$$+ \frac{1}{Re\ h^2}(u_{i+1j} + u_{i-1j} + u_{ij+1} + u_{ij-1} - 4u_{ij}),$$

where

$$F_{i+\frac{1}{2}j} = \frac{u_{i+1j}+u_{ij}}{2h},$$

$$F_{i-\frac{1}{2}j} = \frac{u_{i-1j}+u_{ij}}{2h},$$

$$F_{ij+\frac{1}{2}} = \frac{v_{ij+1}+v_{ij}}{2h},$$

$$F_{ij-\frac{1}{2}} = \frac{v_{ij-1}+v_{ij}}{2h}.$$

If we assume the approximation to, say, $u_{i+\frac{1}{2}j}$ to be of the form

$$u_{i+\frac{1}{2}j} = \sigma_1 u_{i+2j} + \sigma_2 u_{i+1j} + \sigma_3 u_{ij} + \sigma_4 u_{i-1j}, \tag{3.4}$$

where the σ's are the coefficients of the interpolating polynomial. We then get

$$a_{i+2j}u_{i+2j} + a_{i+1j}u_{i+1j} + a_{i-1j}u_{i-1j} + a_{i-2j}u_{i-2j} + a_{ij+2}u_{ij+2} + a_{ij+1}u_{ij+1} + a_{ij-1}u_{ij-1} + a_{ij-2}u_{ij-2}$$

$$+ a_{ij}u_{ij} \qquad + \frac{1}{\rho}\frac{p_{i+1j}-p_{ij}}{h} = 0, \tag{3.5}$$

where, for example,

$$a_{i+2j} = \sigma_1 F_{i+½j}$$

and

$$a_{i+1j} = \sigma_2 F_{i+½j} + \sigma_1 F_{i-½j} - \frac{1}{Reh^2}$$

Finally, the algebraic equation can be written as

$$\sum_{k=i\pm2,i\pm1,i;l=j\pm2,j\pm1,j} A_{kl} u_{kl} + \frac{1}{\rho}\frac{p_{i+1j}-p_{ij}}{h} = 0, \qquad (3.6)$$

where

$$a_{ij} = \sum_{k\neq i,l\neq j} A_{kl.}$$

Following a similar procedure for v, we get a system of algebraic equations,

$$\sum A^{u}_{kl} u_{kl} + \frac{1}{\rho}\frac{p_{i+1j}-p_{ij}}{h} = 0, \qquad (3.7a)$$

$$\sum A^{v}_{kl} + \frac{1}{\rho}\frac{p_{ij+1}-p_{ij}}{h} = 0, \qquad (3.7b)$$

$$\frac{u_{ij}-u_{i-1j}+v_{ij}-v_{ij-1}}{h} = 0. \qquad (3.7c)$$

Having generated such a large system of algebraic equations, the problem has been transformed into one of achieving their fast and accurate solution. On the whole they can be very difficult to solve by direct inversion, because they are both large in number and nonlinear. The object of this chapter will be to discuss the problems associated with the solution of such systems and to outline some of the methods that have been proposed in relation to performing such a task as effectively a possible.

3.2. Solution Techniques

Engineers are constantly on the look-out for ever more accurate solutions to fluid flows of practical interest, and for the solution to increasingly more complicated problems. The latter requirement may be viewed merely as an extension of the former, since the overriding need is for greater accuracy (less error) from the method of

solution. Given a particular fluid flow, there are only two ways of achieving this.

First, one can devise a more accurate discretisation scheme (see Chapter 2), and second, one can reduce the mesh spacing, h. In the previous chapter it was stated that the error associated with a particular method of discretisation is always proportional to h^n where $n \geq 1$. So for a scheme of any order a reduction in h will lead to a reduction in the error (subject to the comments in Chapter 2 concerning the relevance of order of approximation in regions of sharp changes in gradient). Techniques for reducing discretisation error in the former sense have already been dealt with; the latter alternative will be reviewed here.

Reducing h, thereby using more discrete nodes or grid points, has its limitations, the first being the number of data values that can be stored in the core memory of a given computer. For example, a fluids problem in three dimensions with a mesh size of 1/32 (which is only just sufficient for many problems) requires one megabyte of storage for the three velocity components and pressure alone, regardless of any necessary work arrays for residuals or fluxes.

Another point that should be borne in mind is that the number of floating point operations performed in one iterative sweep of a solution technique is directly proportional to the number of nodes, that is to $1/h^d$, where d is the number of dimensions.

Finally, the number of iterations required to achieve a converged solution increases as the number of nodes increases, which can be seen for the different solution techniques listed in Table 3.1. This together with the above point shows that cpu (central processing unit) time is governed by a power law relation of the form $\text{cpu} \propto n^{\beta}$ where $\beta > 1$ (typically about 1.7).

The effect of the first two restrictions outlined above can be reduced, but not eliminated, by adopting efficient algorithms and programming techniques. The third can be analysed more theoretically, and hopefully overcome.

Number of nodes	Number iterations
4^2	20
8^2	83
16^2	151
32^2	180
64^2	243
128^2	698

Table 3.1: Number of iterations for the solution of the lid-driven cavity at Reynolds number of 1 000 with CCCT(α=0) differencing, for the number of nodes shown.

3.3. Rate of Convergence for Traditional Iterative Solvers

Consider Figure 3.1, showing the plot of the residual against iteration number for the Block Implicit Method (see sub-section 3.4.6) applied to the lid-driven cavity problem (Reynolds number 1 000), which will be described more fully later. It can be seen that although the initial convergence rate is rapid it soon slows down, taking a long time to reach full convergence. Figure 3.2 shows a plot of the rate of convergence, $\gamma=R^n/R^{n-1}$, against number of iterations for the same test case, illustrating that while γ is small initially it soon increases and approaches a value of one, implying very slow convergence. The larger the number of finite difference nodes the more pronounced this problem becomes, see Figure 3.3.

Consider now the local behaviour of a typical relaxation technique. The algebraic equations are solved locally such that the errors there are reduced significantly, but this has little effect on the global error. The global or smooth errors (i.e. ones with wavelength $\lambda \gg h$) are only gradually eliminated by successive relaxation sweeps because their variations are not local. In view of this let us reconsider Figure 3.1. It can be seen that initially the error decreases rapidly - corresponding to elimination of the local errors (wavelength $\lambda \approx h$). The slow convergence that emerges later is caused by the relaxation procedure inefficiently attempting to reduce the smooth errors. This inefficiency leads to the increase of γ seen in Figure 3.2. As the number of nodes employed is increased, relaxation becomes localised to ever smaller regions, while any such procedure is required to eliminate errors with a wavelength ever more removed from the local mesh size. This explains the deterioration of convergence in relation to

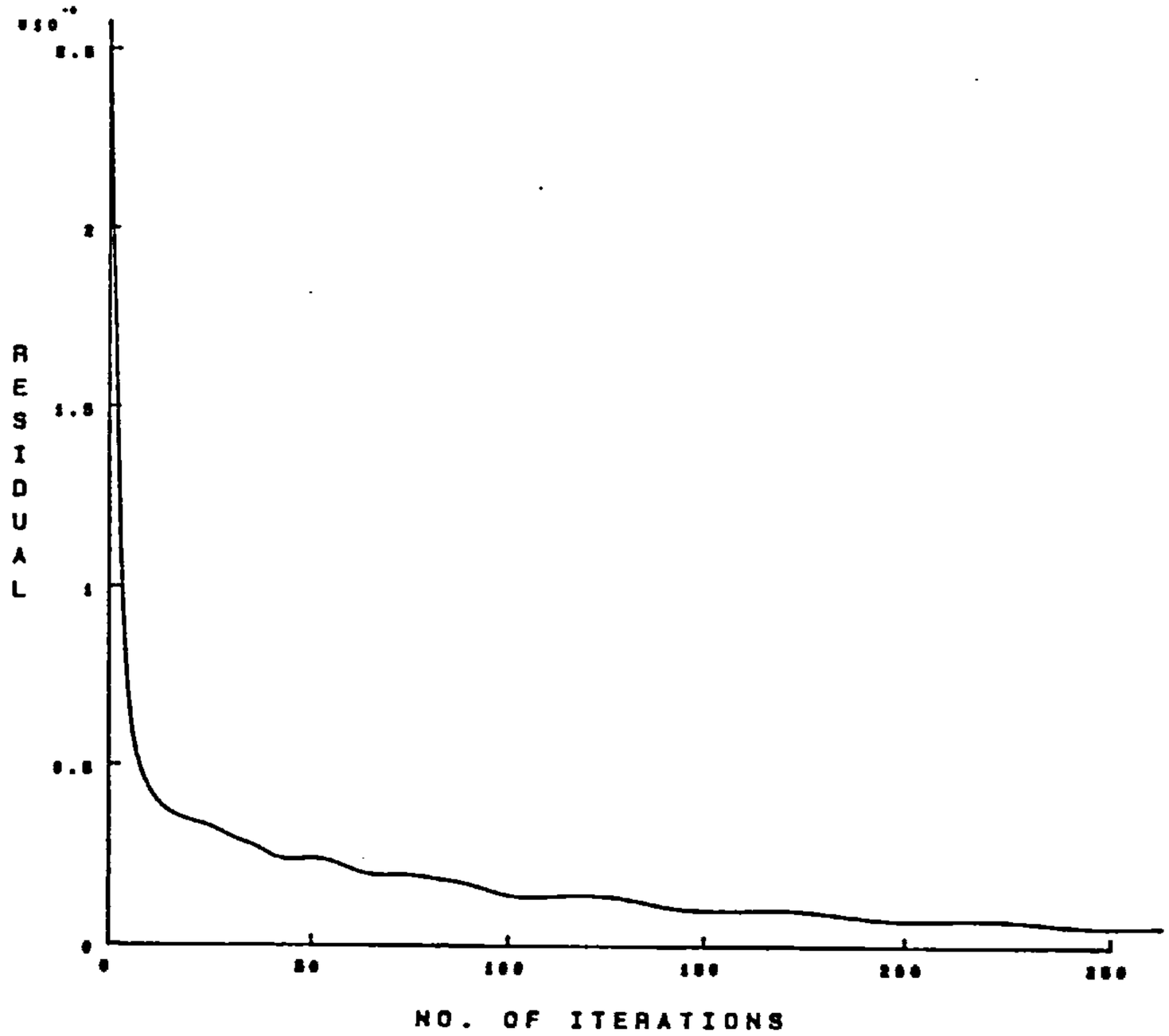


Figure 3.1: Plot of residual against number of iterations obtained on a grid with 128^2 internal nodes and CCCT (α=0) differencing. for Reynolds number of 1000.

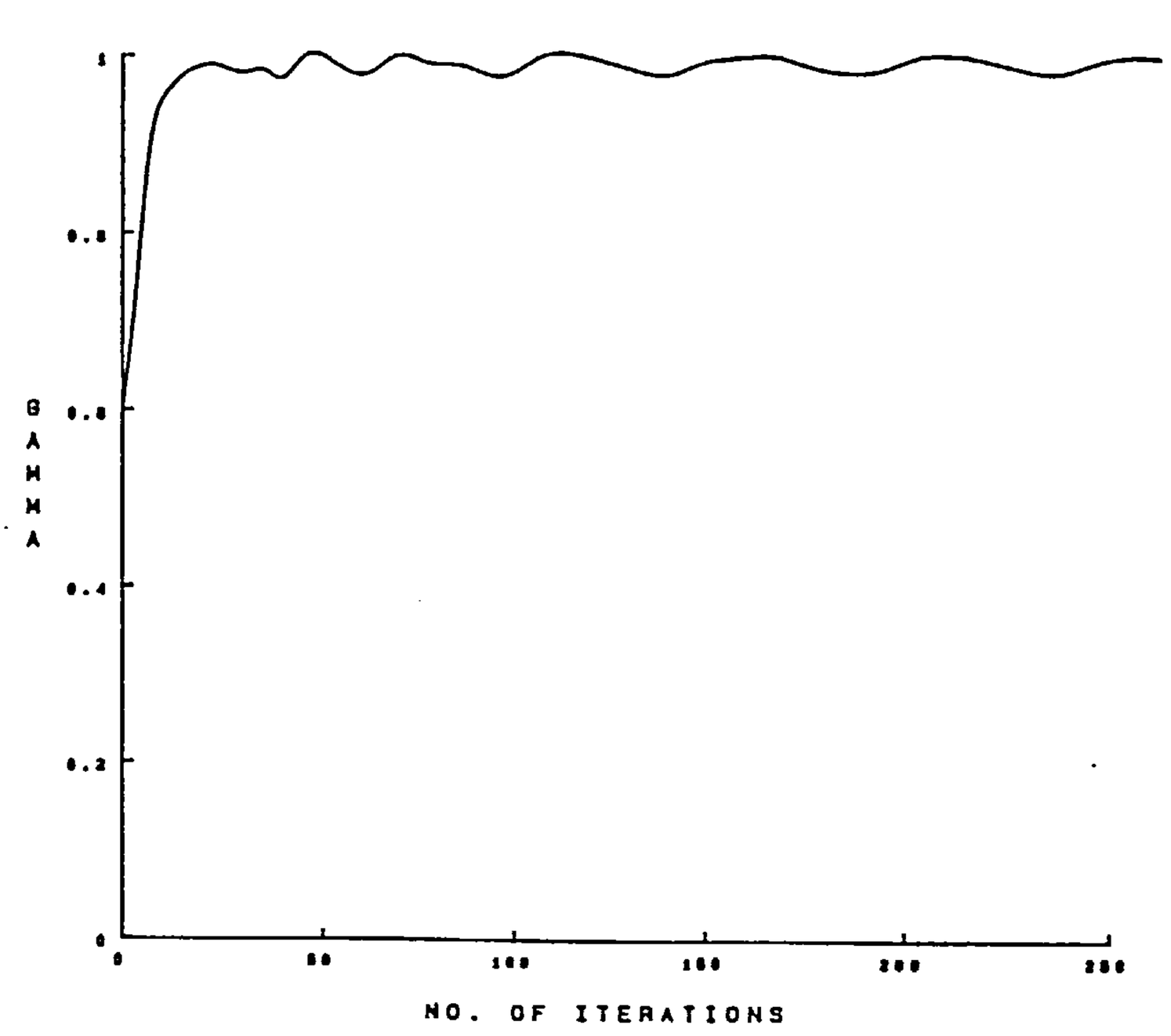


Figure 3.2: Plot of γ against number of iterations for the same case as Figure 3.1.

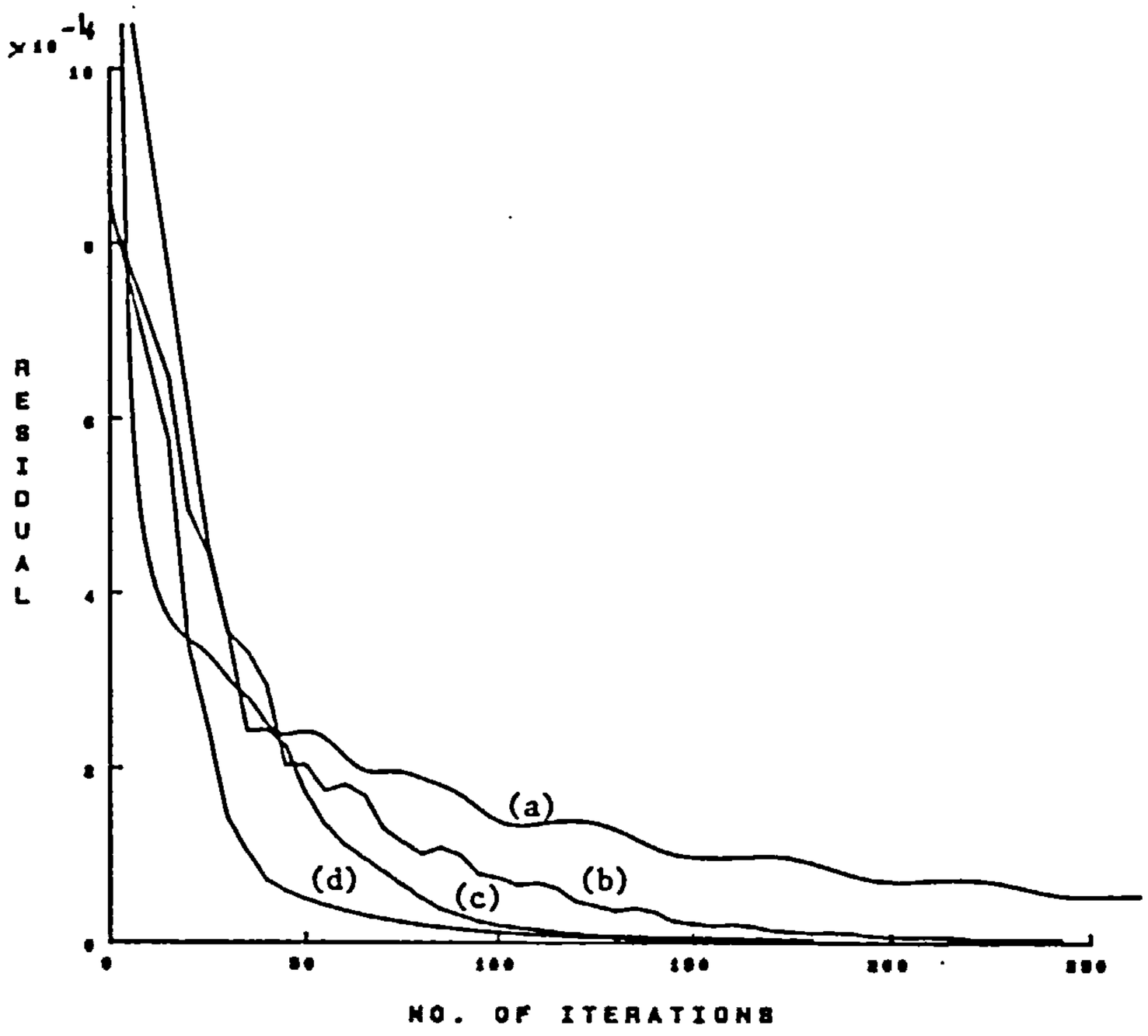


Figure 3.3: Plot of residual against number of iterations obtained with different number of internal nodes as shown: (a) 128^2, (b) 64^2, (c) 32^2 and (d) 16^2.

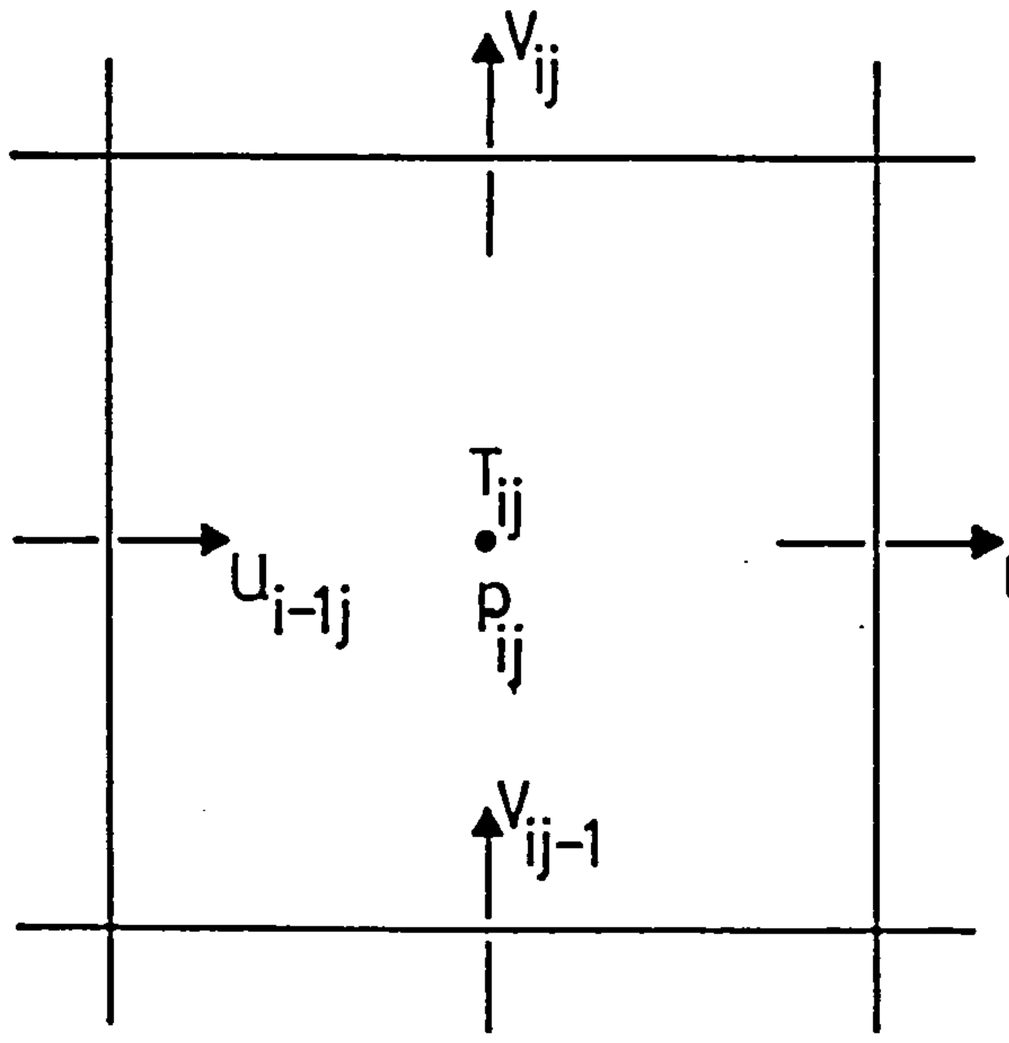


Figure 3.4: Control volume for the Block Implicit Method.

the increase in nodes.

Given that the relaxation scheme is most efficient for errors with wavelengths similar to that of the mesh size, we can consider ways of improving our solution techniques. An error with a wavelength λ that cannot be efficiently eliminated on a particular mesh, may however be successfully handled on a mesh of a different size. Consequently we could use several meshsizes to eliminate all the wavelengths of a given error, optimally. This idea forms the basis of the multigrid technique described more fully in the next chapter. A further advantage to be gained from employing such a procedure would be that relaxation sweeps carried out on a coarser mesh are computationally less expensive than those carried out on finer ones.

These features are attractive and advantageous, and should therefore, if possible, be exploited. However, before constructing and implementing a multigrid technique one must select a suitable relaxation scheme or "smoother" (- so called because we only wish to smooth the error on a given grid, not eliminate it).

3.4. Smoothing Techniques

In order to simulate flows of practical interest the non-linear equations describing the motion are linearised in some way so as to allow updates to the solution to be computed. The way that a smoother deals with this linearisation and the solution locally, is very important - a poor choice can partially negate the benefits that can be accrued with a multigrid algorithm. If it fails to adequately cope with non-linearities, such difficulties may become pronounced at high Reynolds numbers.

Many systems of algebraic equations have a one-to-one equivalence of equations to unknowns. The relaxation scheme is usually based around this equivalence. Each discrete equation is satisfied by changing one unknown - for the elliptic system under consideration here, one can use the u-momentum equation to update u, the v-momentum for v, etc.. When proceeding to the continuity equation it would appear natural to employ it as a means of updating the pressure, especially since pressure nodes are positioned at the same places as those where mass continuity is calculated

when a staggered grid arrangement is used. However, pressure does not appear directly in the continuity equation and so this is not possible. It is important to remember that it is the system of equations as a **whole** that is elliptic, not the individual equations themselves and therefore ideally, the system should be treated as a whole.

There are two alternative approaches to solving the above problem; either the equations of motion are solved simultaneously as a coupled set, or the system can be decoupled and a derived pressure equation employed to determine the pressure field. The most popular option to date has been the latter, which represents the basic idea behind the extremely successful and widely used SIMPLE algorithm devised by Spalding and Patankar[16], but unsegregated solvers are beginning to play a more prominent role. Since the choice of smoother is central to the solution strategy as a whole, some of the more important ones that the are available in the literature are described below. Where appropriate, their use in conjunction with a multigrid algorithm is also discussed.

3.4.1. The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE)

Equation (3.7a) can be rewritten as

$$A_{ij}^{u}u_{ij}=\sum_{k\neq i;l\neq j}A_{kl}^{u}u_{kl}+a_{ij}(p_{i+1j}-p_{ij}). \tag{3.8}$$

(The negative sign has been removed by writing $A_{kl}=-A_{kl}$.)

The first step in the procedure is to assume a pressure field p^* which, when substituted into equation (3.8), gives the associated velocity field u^*,

$$A_{ij}^{u}u_{ij}^{*}=\sum_{k\neq i;l\neq j}A_{kl}^{u}u_{kl}^{*}+a_{ij}(p_{i+1j}^{*}-p_{ij}^{*}). \tag{3.9}$$

Obviously, the u^* velocity field will not exactly satisfy the continuity equation, unless the exact pressure field has been used in equation (3.9). Accordingly, the velocity and pressure fields will need to be corrected;

$$u=u^{*}+u', \tag{3.10}$$

$$p=p^{*}+p'. \tag{3.11}$$

Subtracting (3.9) from (3.8) gives

$$A_{ij}^{u}u_{ij}'=\sum A_{kl}^{u}u_{kl}'+a_{ij}(p_{i+1j}'-p_{ij}'). \tag{3.12}$$

This equation together with equations (3.7c) and (3.10) form a complex system. Patankar and Spalding found that neglecting the first term in the right hand side of equation (3.12) gave a simplified expression which resulted in a more economic computational procedure. So

$$u_{ij}'=D_{ij}(p_{ij}'-p_{i+1j}'), \tag{3.13}$$

where

$$D_{ij}=\frac{a_{ij}}{A_{ij}}.$$

Equation (3.10) now becomes

$$u_{ij}=u_{ij}^{*}+D_{ij}(p_{ij}'-p_{i+1j}'). \tag{3.14}$$

Introducing this equation and corresponding forms for $u_{i-1j}, v_{ij}, v_{ij-1}$ into equation (3.7c) gives,

$$a_{ij}p_{ij}'=\sum_{k=i\pm1;l=j\pm1} A_{kl}P_{kl}'+b, \tag{3.15}$$

with

$$a_{ij}=\sum_{k=i\pm1;l=j\pm1} a_{kl}$$

and

$$b=u_{i-1j}^{*}-u_{ij}^{*}+v_{ij-1}^{*}-v_{ij}^{*}$$

Equation (3.15) is used to update the velocities via equation (3.13) and pressure via equation (3.11). This new pressure field is used to repeat the process until convergence is achieved.

The popularity of the above method has lead to the development of several alternative formulations such as SIMPLER[14,15] and SIMPLEC[4]. However, it is interesting to note that Chleboun[3] reports that as the problem under consideration becomes more complex the difference in the overall performance of these becomes less apparent. This has also been observed by Jones[12].

The one major drawback of this technique is that it decouples the momentum and continuity equations, such that each one is solved sequentially. Thus, no account is taken of changes in the 'other' variables whilst iterating. These 'other' variables are in effect 'frozen'. This evidently affects the coupling, and thus the effectiveness of the solution technique as a whole, in treating non-linearities in the algebraic equations. It has become common practice to solve for each variable in a line-by-line alternating direction fashion, which tends to cope with the non-linearities better than, say, a point-by-point method[9].

The widespread adoption of this approach in industrial and academic spheres means that it obviously cannot be ignored from the point of view of marrying it with a multigrid algorithm. In fact prompted by an examination of the Rolls Royce PACE program, Shaw and Sivaloganathan[19,20] made a thorough investigation of its suitability in such a role. This is discussed in more detail in Chapter 4.

3.4.2. The Distributive Gauss-Seidel Approach (DGS)

In view of the ellipticity of the system and the problem of p not appearing in the continuity equation, Brandt[2] proposed that, following relaxation of the velocities in the normal way (i.e., using the momentum equations), one should use a 'distributive' relaxation for p. Such an approach changes the velocity and pressure values at several nodes in the vicinity of the current point, so as to satisfy the continuity equation, without changing the residuals of the remaining equations in the system.

Let

$$r_{ij}^{c} = \frac{(u_{i+½j} - u_{i-½j} + v_{i+½j} - v_{i-½j})}{h} \tag{3.16}$$

be the continuity residual at (x_i,y_j) before relaxing there.

The relaxation step consists of the following 9 steps;

$$u_{i+½j} = u_{i+½j} + \delta$$

$$u_{i-½j} = u_{i-½j} - \delta$$

$$v_{i+½j} = v_{i+½j} + \delta$$

$$v_{i-½j} = v_{i-½j} - \delta$$

$$p_{ij} = p_{ij} + \frac{4\delta}{h} \tag{3.17}$$

$$p_{i+1j} = p_{i+1j} - \frac{\delta}{h}$$

$$p_{i-1j} = p_{i-1j} - \frac{\delta}{h}$$

$$p_{ij+1} = p_{ij+1} - \frac{\delta}{h}$$

$$p_{ij-1} = p_{ij-1} - \frac{\delta}{h}$$

$$\text{and} \quad \delta = \frac{h}{4} r_c^{ij}.$$

So the continuity residual disappears and the momentum residuals are unchanged. However, the problems inherent in SIMPLE are still present.

Each of the candidate smoothers outlined above is seen to update the variables sequentially, and so represents 'decoupled' or 'segregated' solution techniques which can lead to slower solution times, and in some cases even divergence[18].

A preferable approach would be one that solved the equations directly, thus requiring no iterative procedure. However, as mentioned earlier, this is impractical

from both processing time and storage capacity considerations, but several attempts have been made to develop a solver that treats the variables in an unsegregated iterative manner.

3.4.3. A Coupled Equation Line Solver - (CELS)

This solution technique, proposed by Galpin, van Doormaal and Raithby[5] in 1985, solves the conservation equations in their original form , and so requires no derived pressure correction equation. The method proceeds line-by-line, the solution domain being swept in each co-ordinate direction separately until convergence is obtained. On each line all the equations (momentum and mass conservation) are solved at once and all the variables (u_i,p) are updated simultaneously.

CELS differs from the SIMPLE technique in which a sweep of the whole domain is carried out for one variable before proceeding to the next. Galpin et al found CELS to be more stable and efficient for the test cases that they considered, features that can be explained in terms of the superior coupling of the equations. Galpin and Raithby[6] have also suggested an extension of this method for the treatment of thermal flows.

Consider the equations along the jth line,

$$A^u_{i-1}u'_{i-1} + A^u_i u'_i + A^u_{i+1}u'_{i+1} + \frac{p'_{i+1}-p'_i}{h} = r^u_i, \tag{3.18}$$

$$A^v_{i-1}v'_{i-1} + A^v_i v'_i + A^v_{i+1}v'_{i+1} + \frac{p'_i}{h} = r^v_i, \tag{3.19}$$

$$\frac{u'_i-u_{i-1}'}{h} + \frac{v'_i}{h} = r^c_i. \tag{3.20}$$

Equation (3.20) gives an expression for v'_i, which is used in equation (3.19) to yield an expression for p_i, which in turn is substituted into equation (3.18). This can then be solved to give the u_i's along the line, followed by the v_i's from equation (3.20) and then the p_i's from equation (3.19).

However, the method requires two special adaptations for stability. First, the pressure update must be amended to account for the lack of coupling in the transverse direction. Second, the solution method on the last line must be adjusted to account for the boundary conditions and the difference between the number of unknowns and the number of equations for each variable. The latter problem represents an undesirable constraint, and this together with it's rather complex overall nature effectively ruled it out as an appropriate solver. This is not to say however that it does not have a promising future, if these problems can be overcome, perhaps by adopting a different grid arrangement (see Chapter 7).

3.4.4. A Direct Banded Simultaneous Variable Solution (DBSVS)

Zedan and Schneider[23] carried out an investigation into unsegregated solution strategies and proposed a technique of their own. They considered the two momentum equations;

$$A^{uu}u + A^{up}p = F^{u}, \tag{3.21}$$

$$A^{vv}v + A^{vp}p = F^{v}, \tag{3.22}$$

which give

$$u = -(A^{uu})^{-1}A^{up}p + (A^{uu})^{-1}F^{u}, \tag{3.23}$$

$$v = -(A^{vv})^{-1}A^{vp}p + (A^{vv})^{-1}F^{v}. \tag{2.24}$$

Here A^{uu} is a pentadiagonal matrix and $(A^{uu})^{-1}A^{up}$ can be found using forward and backward substitution. These expressions can be used in the continuity equation to generate an equation for pressure. Having found this pressure the velocities can be calculated from equations (3.21) and (3.22).

In order to investigate the properties of this direct technique, Zedan and Schneider assembled the coefficient matrices for a simple, small test problem and examined the velocity-pressure coupling. They found that the coupling was strongest in the region immediately surrounding the current node.

In view of this, they suggested that when assembling the necessary matrices only the coefficients associated with nearby nodes be used to calculate the coefficients in $(A^{uu})^{-1}A^{up}$ etcetera. The resulting matrix for p - A^{pp} - is found to have a thirteen point structure, and is solved using the Strongly Implicit Procedure (SIP) - proposed by Jacobs[11] for biharmonic problems . An iterative process is used to accommodate the approximation. This solution technique is called Approximate Effect Simultaneous Variable Solution (AESVS).

Later, Zedan and Schneider[24] suggested a further variant named the Strongly Implicit Simultaneous Variable Solution (SISVS). This is similar to AESVS, but has fewer coefficients in the matrices, uses pentadiagonal and bidiagonal forms and implements a partial cancellation parameter in the SIP. They applied this method to the well-known two dimensional lid-driven cavity problem, for Reynolds numbers of 100, 400 and 1 000, on a mesh of 10^2, using Skew Upwind Differencing[17] for the convective term. The results appear to be good, but no comparisons were given with other techniques.

The extension of this technique to large grids would probably make the associated matrices unmanageable, and the use of higher order approximations to convection, such as CCCT differencing[8] would increase the number of non-zero coefficients quite considerably. However, this work does give some insight into the behaviour of unsegregated solution techniques.

3.4.5. A Block Implicit Algorithm using Newton's Method

Recently Vanka[21] implemented an algorithm which makes use of Newton's method. The problem is formulated as follows;

$$F(X)=0 \tag{3.25}$$

$$X=(X_{11},X_{12},.....)^T$$

$$F=(F_{11},F_{12},.....)^T$$

where

$$X_{ij}=(u_{ij},v_{ij},p_{ij})^T,$$

$$F_{ij}=(F^u_{ij},F^v_{ij},F^c_{ij})^T.$$

A Newton Method is then used to solve equation (3.25),

$$X_n = X_{n-1} - \left[\frac{\partial F}{\partial X}\right]^{-1}_{n-1} F_{n-1}, \tag{3.26}$$

or

$$\left[\frac{\partial F}{\partial X}\right]_{n-1} X_{n-1}=F_{n-1}, \tag{3.27}$$

The problem of performing differentiation for all variables is eased by the use of exponential differencing[14], but is still rather complex. The linear equation (3.27) is solved using a direct technique.

This algorithm is obviously complicated to implement and requires large amounts of storage. If a discretisation technique other than the exponential is used, the differentiation of F is even more problematic. To overcome the storage problem, Vanka suggested alternative ordering of the grid points (to make the Jacobian matrix more sparse) and the splitting of the domain into smaller subdomains for separate calculation. However, this method was not used because of its impracticality.

3.4.6. The Block Implicit Method (BIM)

Vanka[22] subsequently proposed a scheme that is both simple to implement and efficient. He named it Symmetrical Coupled Gauss-Seidel (SCGS), but it may be thought of as a combination of DGS and CGS (Collective Gauss-Seidel - used on an unstaggered mesh when (u_{ij},v_{ij},p_{ij}) are updated simultaneously at each point - usually unstable). With SCGS in two dimensions, the four velocities and one pressure corresponding to one control volume are updated simultaneously by inverting a 5 x 5 matrix. Each control volume is visited in turn. Thus each velocity is updated twice. Vanka observed that this ensured the stability that a single update lacked. He

implemented this scheme with hybrid differencing and multigridding, with good results with mesh sizes of 1/320 and Re=2000.

The details of the implementation adopted here differ in some ways from those of Vanka and are thus outlined in full.

Consider the control volume shown in Figure 3.4. We wish to update the unknowns $\{u_{ij}, u_{i-1j}, v_{ij}, v_{ij-1}, p_{ij}\}$. The equations are

for u_{mn}, m= i & i-1, n=j

$$\sum A^{u}_{kl} u_{kl} + \frac{p_{m+1n} - p_{mn}}{\rho h} = 0, \tag{3.28}$$

for v_{mn}, m= i, n=j & j-1

$$\sum A^{v}_{kl} v_{kl} + \frac{p_{mn+1} - p_{mn}}{\rho h} = 0, \tag{3.29}$$

and

$$\frac{u_{ij} - u_{i-1j}}{h} + \frac{v_{ij} - v_{ij-1}}{h} = 0. \tag{3.30}$$

These can be written in terms of residuals and corrections as

$$A^{u}_{ij} u'_{ij} - \frac{p'_{ij}}{\rho h} = r^{u}_{ij},$$

$$A^{u}_{i-1j} u'_{i-1j} - \frac{p'_{ij}}{\rho h} = r^{u}_{i-1j}, \tag{3.31}$$

$$A^{v}_{ij} v'_{ij} - \frac{p'_{ij}}{\rho h} = r^{v}_{ij},$$

$$A^{v}_{i-1j} v'_{i-1j} - \frac{p'_{ij}}{\rho h} = r^{v}_{i-1j},$$

$$\frac{u'_{ij} - u'_{i+1j}}{h} + \frac{v'_{ij} - v'_{ij-1}}{h} = r_{c},$$

to yield the following matrix system

$$\begin{bmatrix} A^u_{ij} & & & & \frac{1}{h} \\ & A^u_{i-1j} & & & -\frac{1}{h} \\ & & A^v_{ij} & & \frac{1}{h} \\ & & & A^v_{ij-1} & -\frac{1}{h} \\ -\frac{1}{\rho h} & \frac{1}{\rho h} & -\frac{1}{\rho h} & \frac{1}{\rho h} & 0 \end{bmatrix} \begin{bmatrix} u'_{ij} \\ u'_{i-1j} \\ v'_{ij} \\ v'_{ij-1} \\ p'_{ij} \end{bmatrix} = \begin{bmatrix} r^u_{ij} \\ r^u_{i-1j} \\ r^v_{ij} \\ r^v_{ij-1} \\ r^c_{ij} \end{bmatrix}. \tag{3.32}$$

This diagonal doubly bordered sparse matrix can be decomposed into lower and upper diagonal (LU) form, the unknown values being found by forward and back substitution. The procedure is very efficient and is outlined in Appendix II.

As a result of the linearisation involved in the calculation of A^u and A^v from values at the previous iterate, the corrections need to be underrelaxed. The velocities are multiplied by α_u before being applied, and the pressure by α_p.

After carrying out this procedure at (x_i,y_j) we continue to (x_{i+1},y_j), reevaluate the matrix and evaluate a new set of corrections. The solution proceeds in order of increasing i then j.

This method demonstrates simplicity and low operation count. As such it was considered to be the most appropriate technique to employ for evaluating a higher order discretisation multigrid scheme.

3.5. Application of the Block Implicit Method

The problem of the two-dimensional lid-driven cavity was solved using both hybrid and CCCT($\alpha = 0$) for Reynolds numbers 100, 400 and 1 000, with numbers of internal nodes ranging from 4^2 to 128^2. The geometry and boundary conditions are shown in Figure 3.5. The equations and grid used here require no boundary condition for the pressure. This is dealt with by specifying the pressure at a certain point to be zero. So after one iteration, the new value of the pressure at the specified point is subtracted from all the pressure values. In the work presented here this point was taken to be at $x = 0.5$, $y = 0.5$.

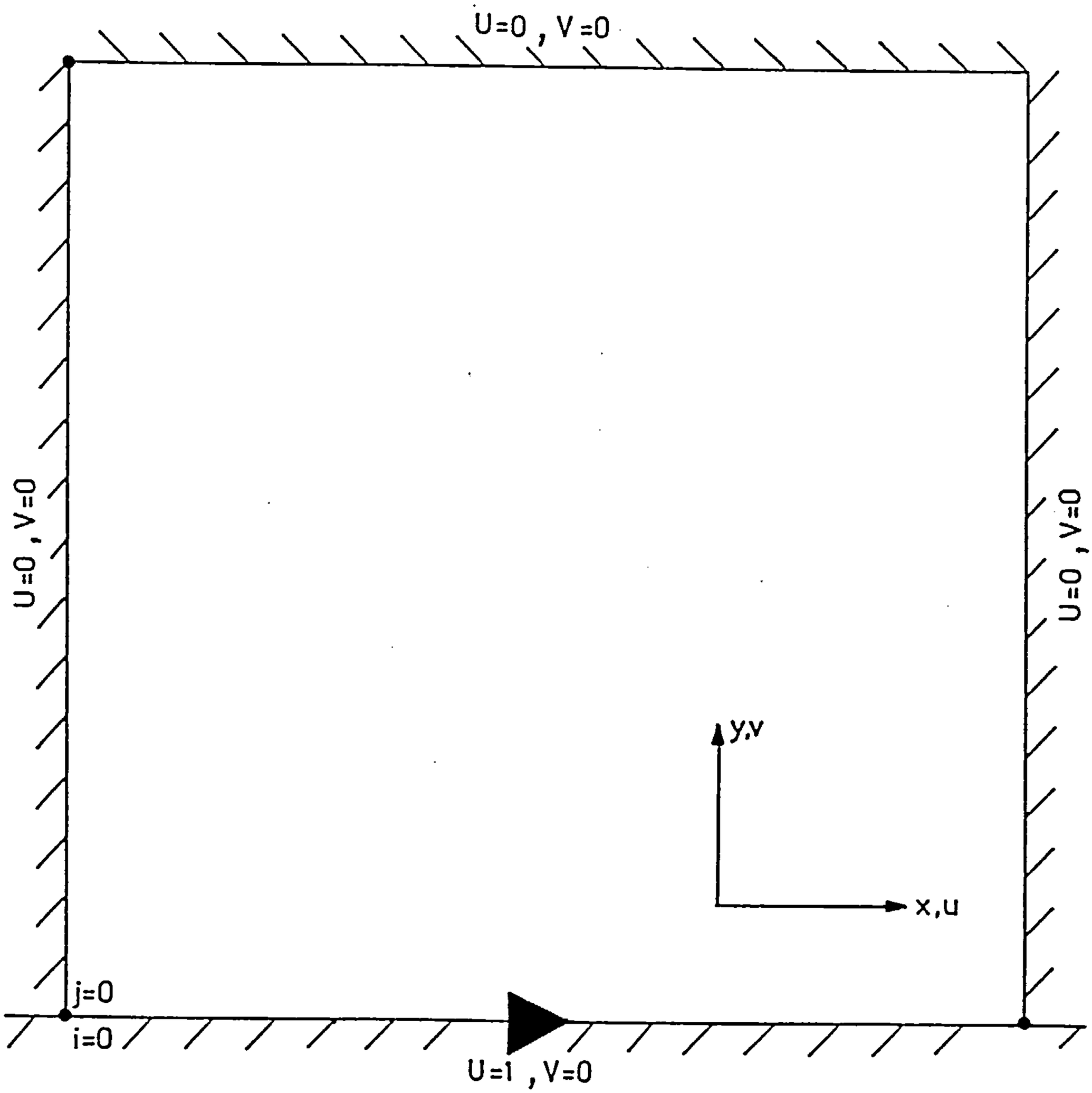


Figure 3.5: Lid-driven cavity flow configuration and boundary conditions.

The iteration proceeds until the residual is less than a prescribed value. Here, this value is taken to be 5×10^{-5} The residual is measured on the basis of the $||.||_2$ norm, i.e.

$$|| r || = \frac{\left(\sum_{i,j} \left(r_u^{ij^2} + r_v^{ij^2} + r_c^{ij^2} \right) \right)^{1/2}}{3 \times n \times n}. \qquad (3.33)$$

Many authors have presented work in this area without giving adequate descriptions of boundary conditions and their implementation. Also, there appears to be several different approaches, and therefore a detailed explanation of the methods used here will be given.

3.5.1. Boundary Conditions

Consider Figure 3.6(a). The boundary nodes are positioned on the boundary at a distance h/2 from the near boundary nodes. In the set of algebraic equations the coefficients of the near boundary values are amended to account for this halved mesh-size.

In the interior of the domain,

$$\frac{\partial}{\partial y}\left[u\phi\right] = \frac{v_{ij}\phi_{ij+1/2} - v_{ij-1}\phi_{ij-1/2}}{h} \qquad (3.34)$$

When using the boundary configuration in Figure 3.6(a), $\phi_{ij-1/2}$ is given directly by ϕ_{ij-1}. If hybrid is used to approximate $\phi_{ij+1/2}$ with ϕ_{ij} and ϕ_{ij+1}, the boundary configuration is of no consequence. If CCCT$(\alpha=0)^8$ is used with ϕ_{ij-1},ϕ_{ij}andϕ_{ij+1} then the interpolating polynomial must be amended to take account of the different mesh spacing.

Also consider $\left[\frac{\partial\phi}{\partial y}\right]_{ij-1/2}$; at the boundary this is approximated as

$$\frac{\phi_{ij} - \phi_{ij-1}}{(h/2)}.$$

An alternative to the above approach was considered; see Figure 3.6(b). Extra points are introduced at 'image' positions a distance h from the near boundary nodes and therefore equidistant from the boundary on either side. These image nodes are fixed so

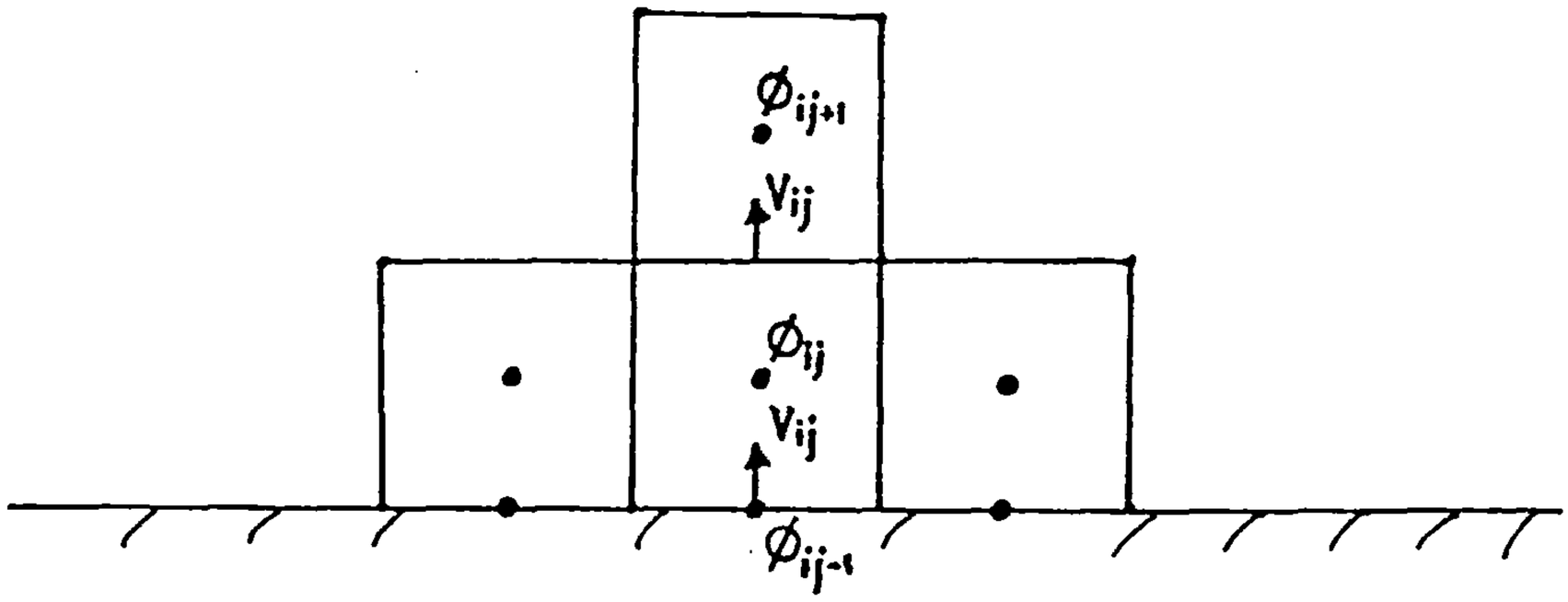


Figure 3.6(a): Configuration of boundary nodes used here.

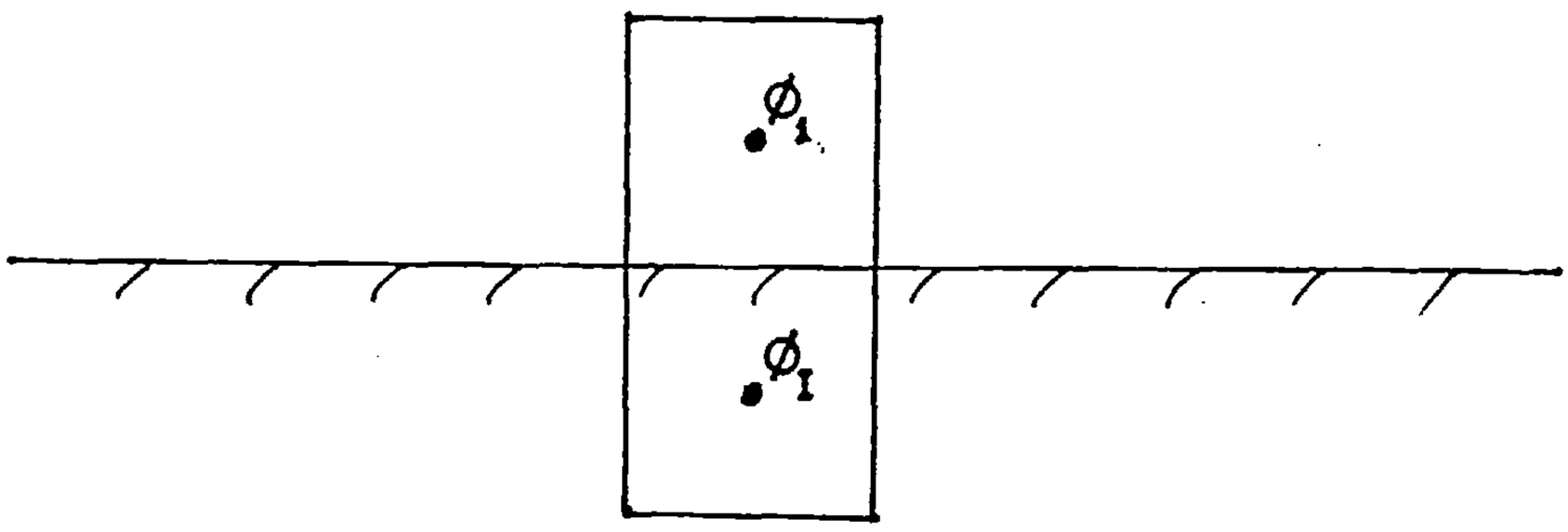


Figure 3.6(b): Configuration of boundary nodes using image points.

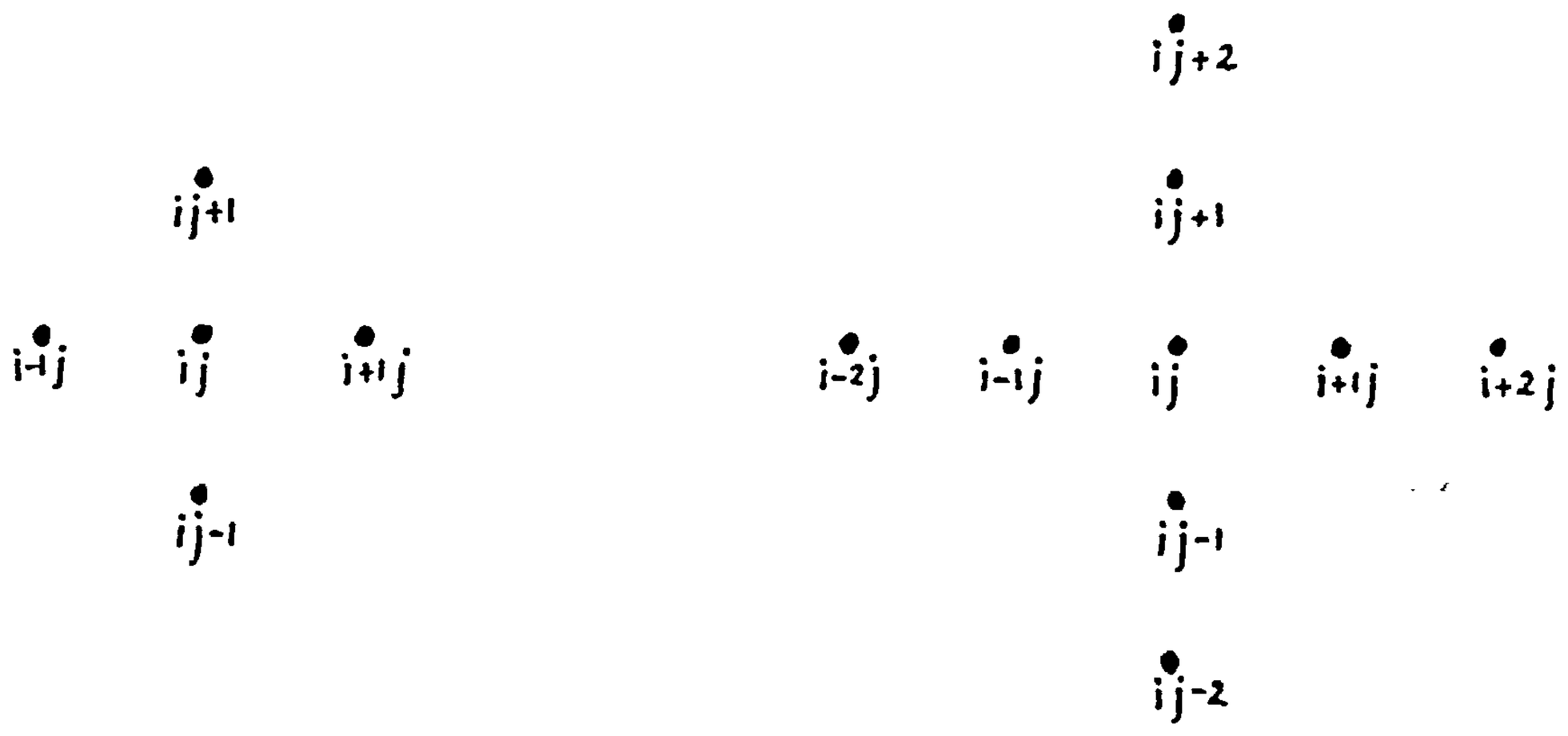


Figure 3.7: Computationed molecules for (a) hybrid and (b) CCCT (α=0) differencing.

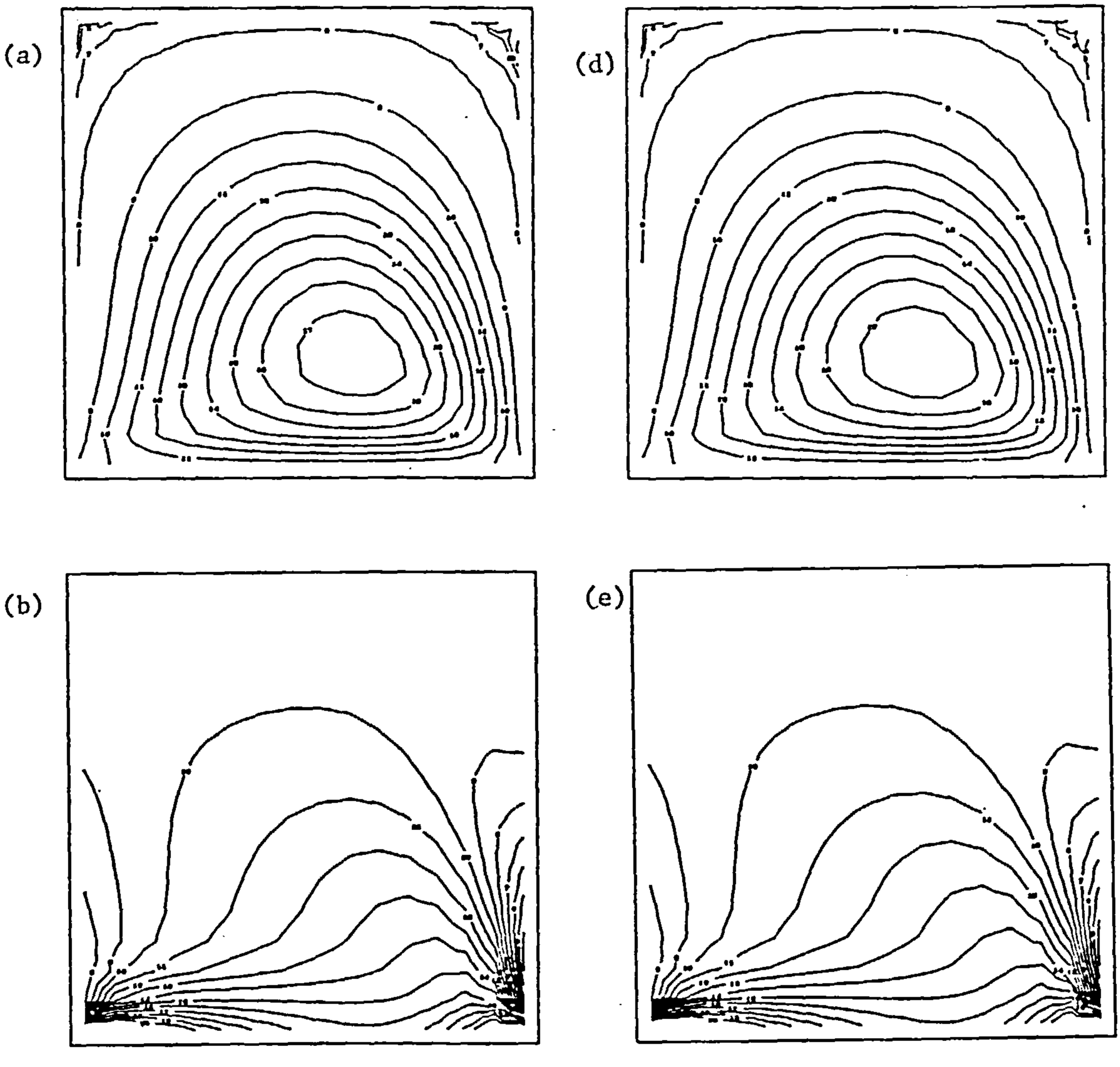


Figure 3.8: Contour plots for Reynolds number 100 with 16^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

as to satisfy the imposed boundary conditions. So, if central differencing is used and the boundary value is zero, ϕ_I is set as $-\phi_I$, or if $\frac{\partial\phi}{\partial y}=0$ then $\phi_I=\phi_I$.

The first technique is common practice[7] and has been successfully applied for a wide range of complex problems. Leonard[13], on the other hand, uses the second technique, but has only reported results for simple test problems. Both techniques were implemented in the present work. Although the second method is easier to program and requires less cpu time per iteration (there are less logical expressions to evaluate), it was seen to require more iterations, so it actually used more cpu time than the first technique. Also at high Reynolds numbers it was often unstable, particularly when using a multigrid technique.

In cases where CCCT discretisation is used, when approximating the near-boundary values of velocity perpendicular to the boundary, use is made of hybrid[16] differencing to calculate the near-boundary flux.

In the problems considered here, there is no boundary condition for pressure. As explained earlier, this was resolved by taking a reference point and subtracting its value there from all points after each iteration. Vanka[22] took this point as one of the corners. Here, instead, the value at the centre was calculated as the average of its four surrounding values and this average was subtracted from all other nodal values. This procedure is just as efficient as Vanka's, but ensures better correspondence between the pressures on grids of a different size.

3.5.2. Sweeping Procedures

In his original work Vanka[22] solved for each control volume one by one in the direction of increasing i then j. However, this is obviously not the only way of proceeding; i or j could decrease or j could change before i. In fact there are eight different possibilities;

(1) i increasing j increasing

(2) i decreasing j increasing

(3) i increasing j decreasing

(4) i decreasing j decreasing

(5) j increasing i increasing

(6) j decreasing i increasing

(7) j increasing i decreasing

(8) j decreasing i decreasing

The relative performance of these is affected by the predominant flow direction, so to choose a particular one would not be meaningful in the general case. A combination of some sort would be preferable. As such, several cases were investigated;

(a) (1) then (4)

(b) (5) then (8)

(c) (1) then (4) then (5) then (8)

(d) (1) then (4) then (5) then (8) then (2) then (3) then (6) then (7)

Of the alternative schemes (c) was observed to be the most successful. It was better at smoothing the errors than (a) or (b), but was not as unwieldy as (d). The latter performs eight sweeps per iteration, which results in an error reduction that is often far greater than required. This can make it inefficient, particularly when using multigridding (see next chapter for more details). In Table 3.2 the comparative cpu times are given for (a) and (c) when solving the lid-driven cavity problem with various mesh spacings for Reynolds numbers of 100 and 1 000.

	(a)		(c)	
	Reynolds no.		Reynolds no.	
Mesh	100	1000	100	1000
4^2	0.06	0.12	0.05	0.15
8^2	0.57	1.51	0.31	2.34
16^2	5.82	12.92	2.03	15.45
32^2	75.25	124.97	16.44	73.26
64^2	936.63	1193.66	200.16	710.44

Table 3.2 : Computer time (secs.) for (a) one sweep and (c) four sweeps.

3.5.3. Results

Table 3.3 shows the values of the relaxation factors (velocity - α_u , pressure - α_p) for both discretisations at the three different Reynolds numbers. Those for hybrid vary only slightly as the Reynolds number increases but far less than is the case with CCCT. Relaxation is necessary because of the linearisation; as Reynolds number increases, the problem becomes more non-linear. Therefore, the relaxation factor must decrease. To explain why this phenomenon is more pronounced with CCCT, we must consider the computational molecule (i.e. the extent of the points involved in calculating an update to, say, u) - see Figure 3.7. From this figure it can be seen that in performing one calculation for the Block Implicit Method, where two velocities are updated implicitly, for hybrid six are not, and for CCCT twelve are not. So for hybrid 1/4 of the points are solved implicitly, but for CCCT only 1/7 of the points are solved implicitly. So CCCT represents a much cruder linearisation (in terms of the solution technique) and therefore requires much lower relaxation factors at higher Reynolds numbers.

a) hybrid

	Reynolds no.		
Mesh	100	400	1000
4^2	0.8,1.3	0.8,1.4	0.6,1.6
8^2	0.8,1.3	0.8,1.4	0.6,1.6
16^2	0.8,1.3	0.8,1.4	0.6,1.6
32^2	0.8,1.3	0.8,1.4	0.6,1.6
64^2	0.8,1.3	0.8,1.4	0.6,1.6
128^2	0.8,1.3	0.8,1.4	0.6,1.6

b) CCCT

	Reynolds no.		
Mesh	100	400	1000
4^2	0.7,1.3	0.5,1.3	0.2,1.6
8^2	0.9,1.3	0.5,1.3	0.2,1.6
16^2	1.1,1.4	0.5,1.5	0.2,1.6
32^2	1.1,1.5	0.6,1.5	0.2,1.6
64^2	1.1,1.5	0.8,1.4	0.3,1.6
128^2	1.1,1.5	0.8,1.4	0.3,1.6

Table 3.3: Relaxation factors α_u, α_p) for hybrid and CCCT(α=0) differencing.

From Table 3.4 it can be seen that for both schemes the cpu time obeys the relationship

$$cpu \; \alpha \; n^{BETA} \tag{3.35}$$

where BETA ≈ 1.6 - this behaviour was discussed earlier on in the chapter. It can be seen that solutions for more than 128^2 nodes are impractical with the method as it stands. For example, even at a Reynolds number of 100 the time required for a solution with 256^2 nodes is of the order of 30 000 cpu seconds.

a) Reynolds no. = 100

	Discretisation	
Mesh	hybrid	CCCT
4^2	0.07	0.05
8^2	0.35	0.35
16^2	3.24	2.33
32^2	37.33	19.40
64^2	449.35	264.23
128^2	5887.91	3541.88

b) Reynolds no. = 400

	Discretisation	
Mesh	hybrid	CCCT
4^2	0.09	0.08
8^2	0.35	0.85
16^2	2.26	5.71
32^2	20.51	37.29
64^2	252.38	279.59
128^2	3342.39	3772.66

c) Reynolds no. = 1000

	Discretisation	
Mesh	hybrid	CCCT
4^2	0.07	0.19
8^2	0.49	3.20
16^2	4.22	23.27
32^2	38.13	112.15
64^2	309.18	611.12
128^2	4232.05	7078.15

Table 3.4: Computer time (secs.) for both discretisation schemes at all Reynolds numbers.

Figure 3.1 shows a graph of residual against iteration number for Re = 1 000 and 128^2 nodes. The residual decreases rapidly at first then convergence slows down. This is also demonstrated in Figure 3.2 of γ against iteration number.

Tables 3.5 and 3.6 and Figures 3.8 to 3.19 show the results for streamfunction, vorticity and pressure. The vorticity is calculated directly from u and v, whilst the streamfunction is calculated as the solution of

$$\nabla^2\psi = -\omega$$

using the Strongly Implicit Procedure[11]. From these it can be seen that grid independence has practically been achieved with 128 nodes. The results for CCCT differencing compare well with those of Ghia, Ghia and Shin[10] (see Table 3.7), who used a coupled strongly - implicit method for a streamfunction-vorticity formulation with a multigrid algorithm for h = 1/256. Their results are very accurate not only because of the small mesh size employed, but also as a result of the fact that the streamfunction and vorticity are calculated directly - not indirectly as in the present work.

Consider Figures 3.8 to 3.19 showing plots of streamfunction, vorticity and pressure at the three Reynolds numbers 100, 400 and 1 000. These results were obtained for meshes of 16^2, 32^2, 64^2 and 128^2 internal nodes with both hybrid and CCCT discretisation.

For a Reynolds number of 100 there is little difference between the relative performance of the two schemes. They both converge (as h decreases) to similar values, that are accurate. CCCT is slightly better at predicting the secondary eddies. It is interesting to note that for n>50 the cell Reynolds number (Rh/L) is always less than two, so the hybrid scheme will reduce to central differencing throughout the flow domain. This explains the accuracy for 64^2 and 128^2 points.

For Reynolds number 400 the differences are more apparent. CCCT gives more accurate results with 16^2 nodes. The contours still differ appreciably for 32^2 nodes. The difference is less visible for 64^2 and 128^2 nodes, but reference to Tables 3.5 and 3.6

a) Reynolds number = 100

Mesh	8^2	16^2	32^2	64^2	128^2
ψ_m	0.08923	0.09841	0.10227	0.10305	0.10301
X_m	0.56250	0.59375	0.60938	0.61719	0.61328
Y_m	0.31250	0.28125	0.26563	0.25781	0.26172
V_m	2.1096	2.7770	3.0715	3.1846	3.1462
ψ_r	-	-1.258E-5	-1.581E-5	-1.348E-5	-1.240E-5
X_r	-	0.90625	0.95313	0.94531	0.94141
Y_r	-	0.96875	0.92188	0.92969	0.94141
V_r	-	-0.03535	-0.03820	-0.03997	-0.03343
ψ_l	-4.793E-5	-7.467E-6	-3.143E-6	-2.376E-6	1.9729E-6
X_l	0.06250	0.03125	0.04688	0.03906	0.03516
Y_l	0.93750	0.96875	0.95313	0.96094	0.96484
V_l	-.08561	-.01651	-.03604	-.02176	-0.01568

b) Reynolds number = 400

Mesh	8^2	16^2	32^2	64^2	128^2
ψ_m	0.08505	0.09960	0.10844	0.11224	0.11308
X_m	0.56250	0.53125	0.54688	0.55469	0.55859
Y_m	0.31250	0.40625	0.39063	0.39844	0.39453
V_m	1.5400	2.2096	2.2341	2.2710	2.290
ψ_r	-	-7.361E-4	-7.238E-4	-6.669E-4	-6.419E-4
X_r	-	0.90625	0.89063	0.88281	0.88672
Y_r	-	0.84375	0.85938	0.88281	0.87891
V_r	-	-0.43772	-0.4857	-0.4255	-0.4242
ψ_l	-8.342E-5	-1.579E-5	-1.931E-5	-1.476E-5	-1.351E-5
X_l	0.06250	0.03125	0.04688	0.05469	0.05078
Y_l	0.93750	0.90625	0.95313	0.96094	0.94922
V_l	-0.11417	-0.08824	-0.04857	-0.04580	-0.06332

c) Reynolds no. = 1000

Mesh	8^2	16^2	32^2	64^2	128^2
ψ_m	0.08161	0.09825	0.10798	0.11430	0.11574
X_m	0.56250	0.53125	0.51563	0.53906	0.53516
Y_m	0.43750	0.40625	0.42188	0.42969	0.43359
V_m	1.6752	1.8119	1.9082	1.9982	2.0402
ψ_r	-	-1.342E-3	-2.134E-3	-1.812E-3	-1.693E--3
X_r	-	0.90625	0.85938	0.86719	0.86328
Y_r	-	0.84375	0.89063	0.88281	0.88672
V_r	-	-1.0957	-.94255	1.0625	-1.0946
ψ_l	-1.339E-4	-2.087E-4	-1.958E-4	-2.197E-4	-2.064E-4
X_l	0.06250	0.09375	0.07813	0.08594	0.08203
Y_l	0.93750	0.90625	0.92188	0.92969	0.92578
V_l	-0.13738	-0.38156	-0.28050	-0.29582	-0.29997

Table 3.5: Streamfunction and vorticity data obtained with CCCT(α=0).

a) Reynolds = 100

Mesh	8^2	16^2	32^2	64^2	128^2
ψ_m	0.08664	0.09710	0.10209	0.10308	0.10317
X_m	0.56250	0.59375	0.60938	0.61719	0.61328
Y_m	0.31250	0.28125	0.26563	0.25781	0.26172
V_m	1.980	2.7199	3.0652	3.1838	3.1470
ψ_r	-	-1.217E-5	-1.551E-5	-1.254E-5	-1.224E-5
X_r	-	0.90625	0.95313	0.94531	0.94141
Y_r	-	0.96875	0.92188	0.94531	0.94141
V_r	-	-.04130	-.03893	-.03987	-0.3331
ψ_l	-4.793E-5	-7.226E-6	-2.800E-6	-2.415E-6	-2.083E-6
X_l	0.06250	0.03125	0.04688	0.03906	0.03516
Y_l	0.93750	0.96875	0.95313	0.92969	0.96484
V_m	-.08561	-.01632	-.03581	-.02186	-0.01584

b) Reynolds no. = 400

Mesh	8^2	16^2	32^2	64^2	128^2
ψ_m	0.06618	0.08140	0.09991	0.11103	0.11306
X_m	0.56250	0.59375	0.57813	0.55469	0.55859
Y_m	0.18750	0.34375	0.39063	0.39844	0.39453
V_m	3.9330	1.6886	1.6702	2.0783	2.2492
ψ_r	-	-1.899E-4	-5.447E-4	-6.335E-4	-6.414E-4
X_r	-	0.90625	0.89063	0.88281	0.88672
Y_r	-	0.90625	0.89063	0.88281	0.87891
V_r	-	-0.29438	-0.33968	-0.43227	-0.42802
ψ_l	-6.041E-5	-8.351E-6	-1.469E-5	-1.458E-5	-1.354E-5
X_l	0.06250	0.03125	0.04688	0.03906	0.05078
Y_l	0.93750	0.96875	0.95313	0.94531	0.94922
V_l	-.10564	-.01447	-.04967	-.04965	-0.06352

c) Reynolds no. = 1000

Mesh	8^2	16^2	32^2	64^2	128^2
ψ_m	0.05473	0.06307	0.08225	0.10170	0.11516
X_m	0.56250	0.59375	0.54688	0.53906	0.53516
Y_m	0.18750	0.40625	0.42188	0.42969	0.43359
V_m	1.0900	1.0468	1.4223	1.7171	1.9907
ψ_r	-	-1.127E-4	-1.696E-3	-1.812E-3	-1.7427E-3
X_r	-	0.90625	0.89063	0.86719	0.86328
Y_r	-	0.96875	0.89063	0.88281	0.88672
V_r	-	-.31862	-1.0852	-1.0625	-1.1442
ψ_l	-9.422E-5	-1.524E-4	-1.539E-4	-2.197E-4	-2.211E-4
X_l	0.06250	0.09375	0.07813	0.08594	0.08203
Y_l	0.93750	0.90625	0.92188	0.92969	0.92578
V_l	-0.16640	-0.33400	-0.22223	-0.29582	-0.31167

Table 3.6: Streamfunction and vorticity data obtained with hybrid

	Re=100		Re=400		Re=1000	
	Present	G,G&S	Present	G,G&S	Present	G,G&S
ψ_m	0.10301	0.10342	0.11308	0.11391	0.11574	0.11793
X_m	0.6133	0.6172	0.5586	0.5547	0.5352	0.5313
Y_m	0.2617	0.2656	0.3945	0.3945	0.4336	0.9438
V_m	3.1462	3.1665	2.2900	2.2947	2.0402	2.0497
ψ_r	-1.24E-5	-1.25E-5	-6.42E-4	-6.42E-5	-1.69E-2	-1.75E-3
X_r	0.9414	0.9453	0.8867	0.8906	0.8633	0.8594
Y_r	0.9414	0.9375	0.8789	0.8750	0.8867	0.8906
V_r	-0.03343	-0.03307	0.42420	-0.43352	-1.0946	1.1547
ψ_l	-1.97E-6	-1.75E-6	-1.35E-5	-1.42E-5	-2.06E-4	-2.31E-4
X_l	0.0352	0.0313	0.0508	0.0508	0.0820	0.0859
Y_l	0.9648	0.9609	0.9492	0.9531	0.9258	0.9220
V_l	-0.01568	-0.01556	-0.06332	-0.05697	-0.29997	-0.36175

Table 3.7: Comparison of results obtained with CCCT(α=0) using 128^2 internal nodes with those of Ghia, Ghia and Shin.

show that CCCT is superior. For n=128^2 the maximum cell Reynolds number is 3.125, so with hybrid, central differencing will be used predominantly, in the hybrid case.

For Reynolds number 1 000 there is a significant difference between the two schemes. The effect of the numerical diffusion introduced by the hybrid scheme can be observed in both the streamfunction and vorticity plots. In particular, compare Figures 3.19 (b) and (e). Even with 128^2 nodes, hybrid underpredicts ψ_M by 10%, whereas CCCT underpredicts it by only 1%.

Although the CCCT scheme usually takes longer to converge for a given number of nodes, it generates much more accurate results, at lower mesh densities. Taking the results of Ghia, Ghia and Shin as a benchmark, we can perform an analysis of accuracy in relation to cpu time (see Figure 3.20). It can be seen that for Reynolds number 1 000, CCCT can generate an answer to a given accuracy in just a fraction of the time required with hybrid.

These results are interesting in themselves, but also serve to illustrate the suitably of the BIM as a smoother for use with a multigrid technique - see Chapter 4.

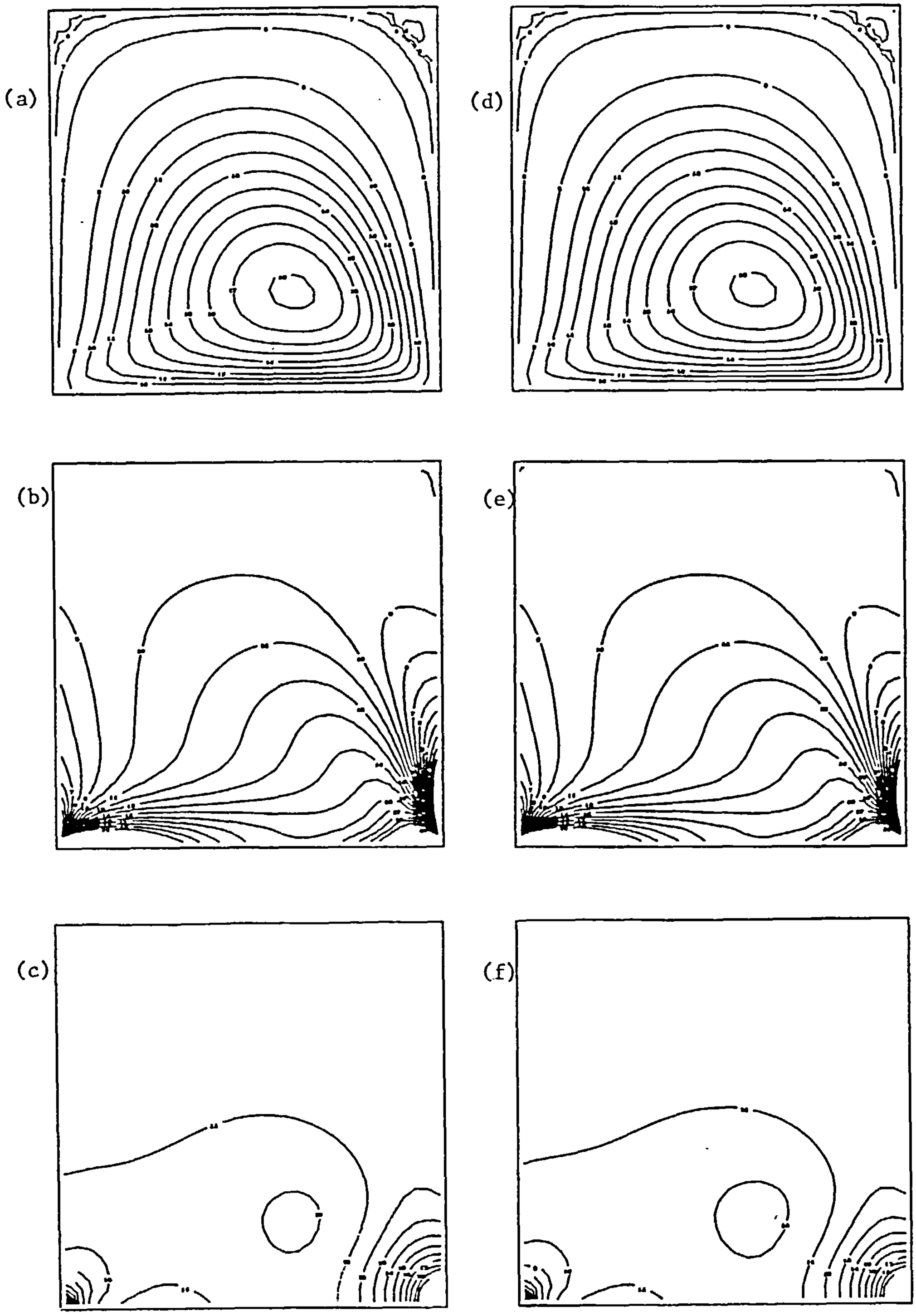


Figure 3.9: Contour plots for Reynolds number 100 with 32^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

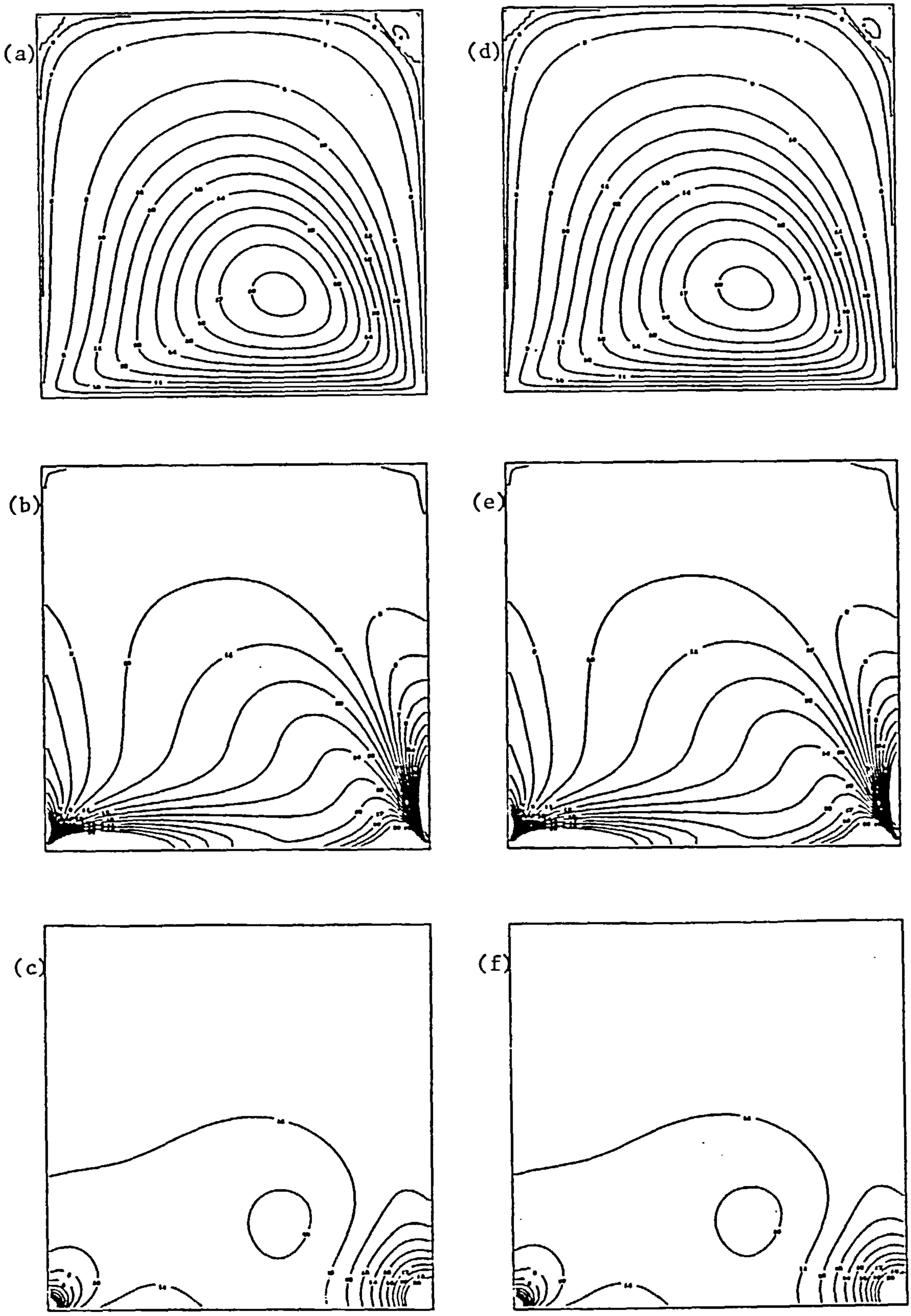


Figure 3.10: Contour plots for Reynolds number 100 with 64^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e): streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

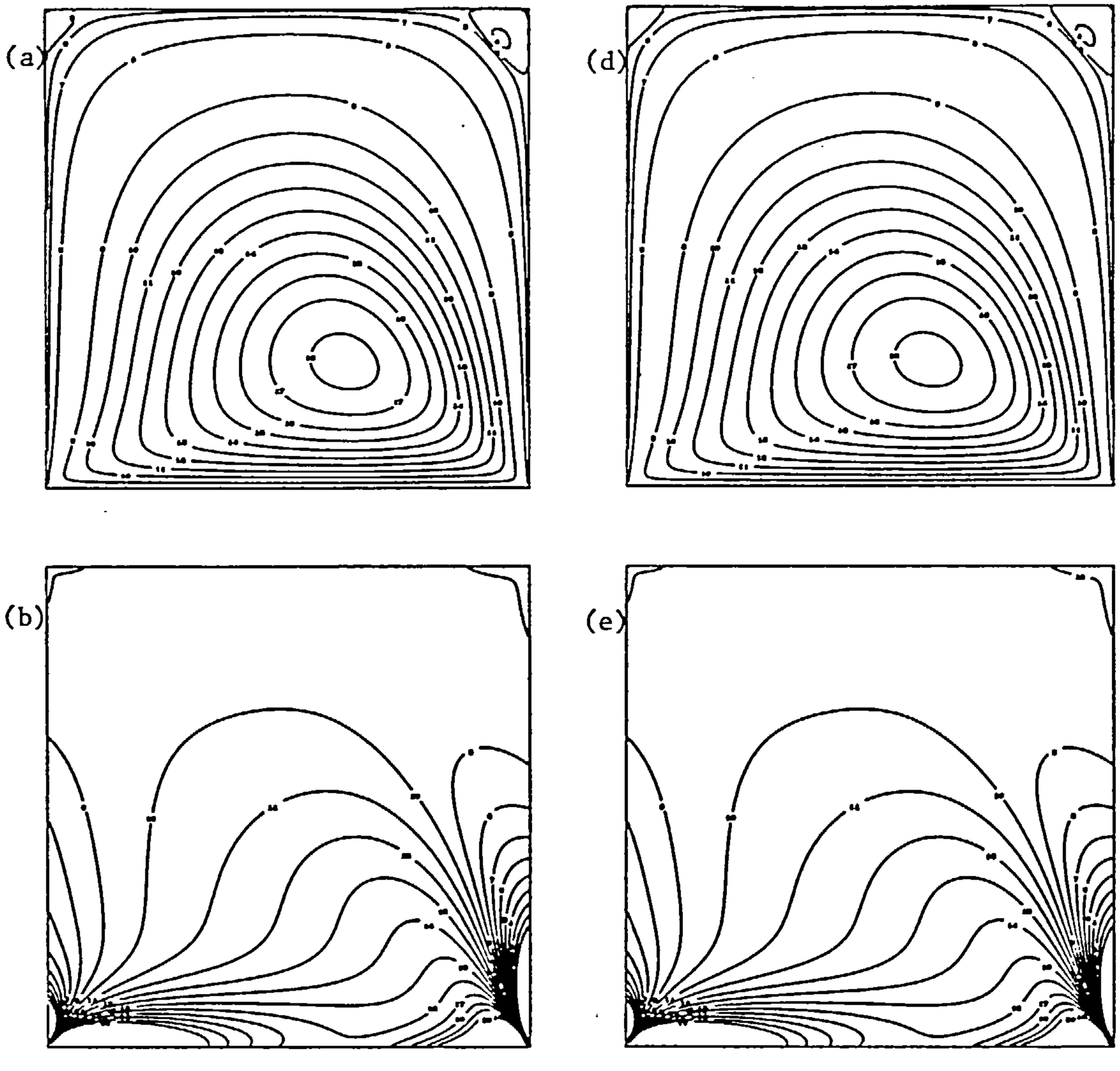


Figure 3.11: Contour plots for Reynolds number 100 with 128^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

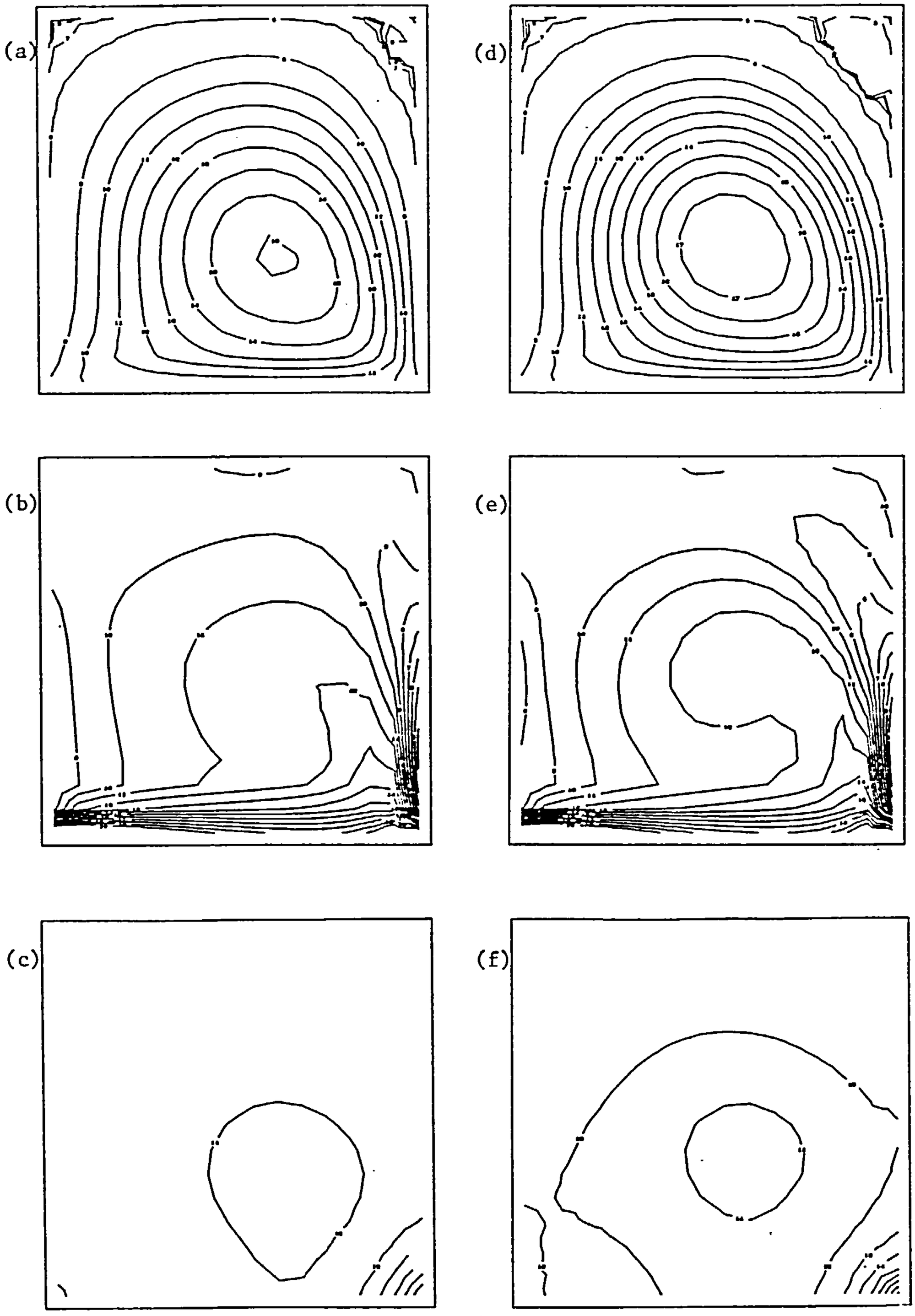


Figure 3.12: Contour plots for Reynolds number 400 with 16^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

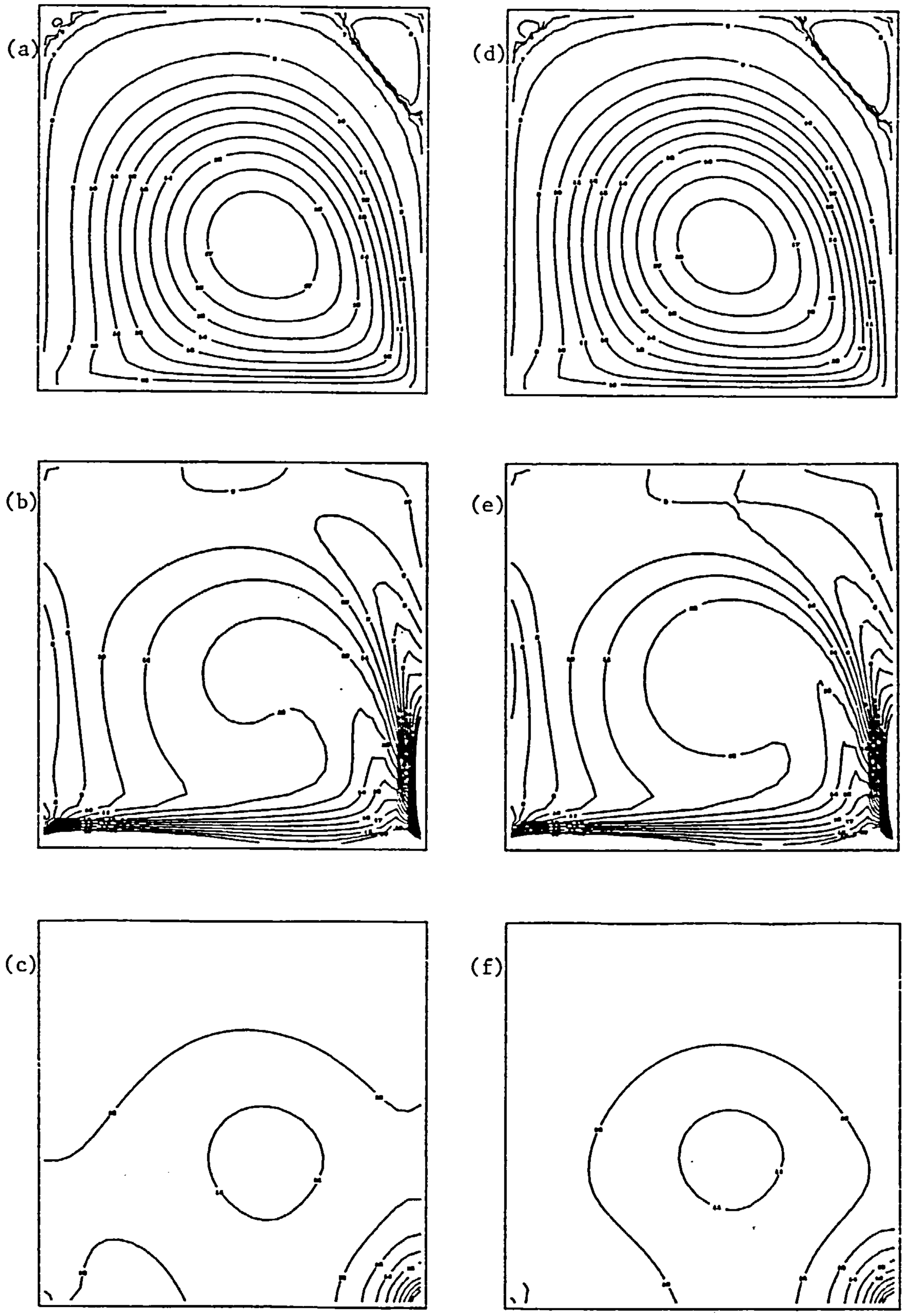


Figure 3.13: Contour plots for Reynolds number 400 with 32^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

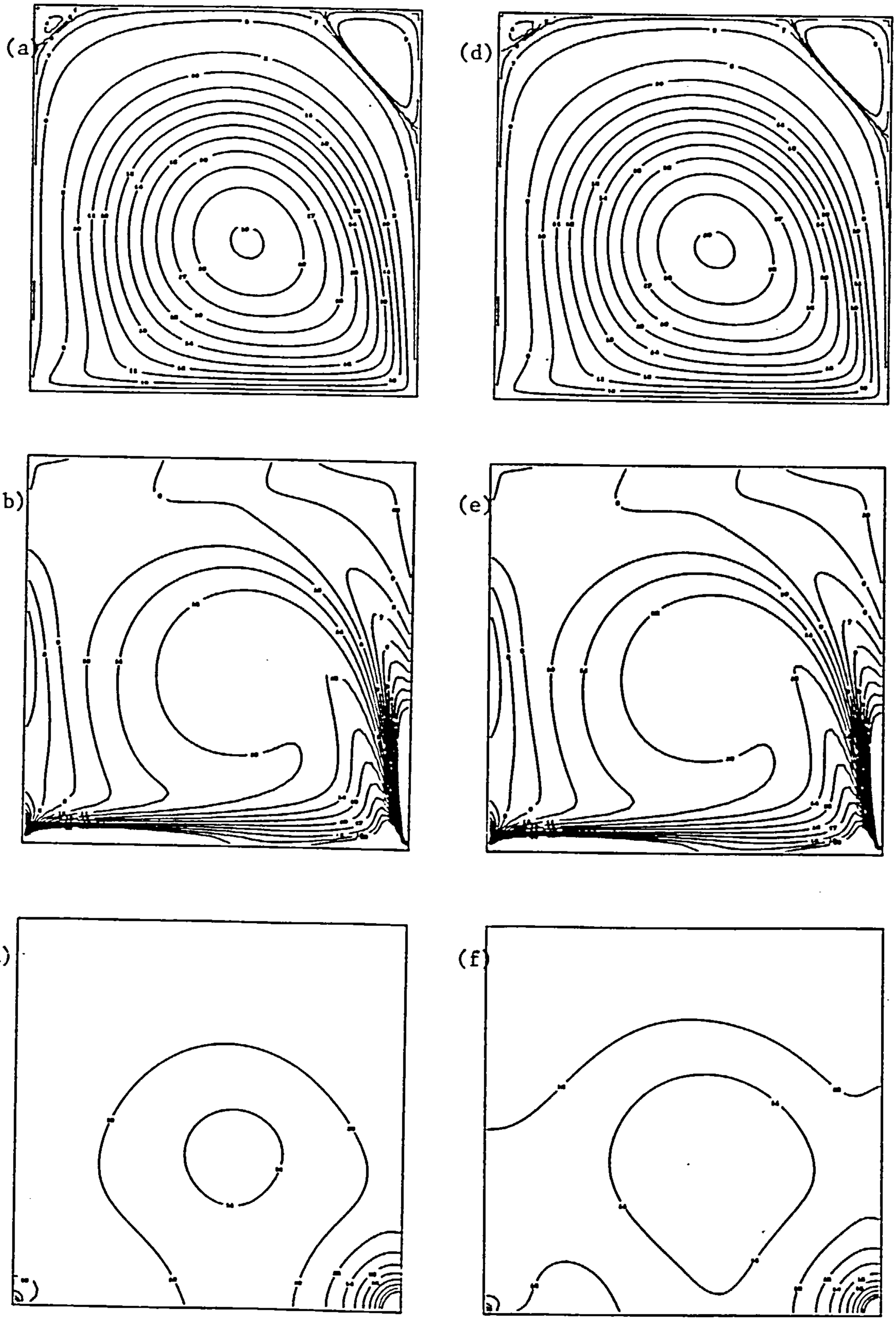


Figure 3.14: Contour plots for Reynolds number 400 with 64^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

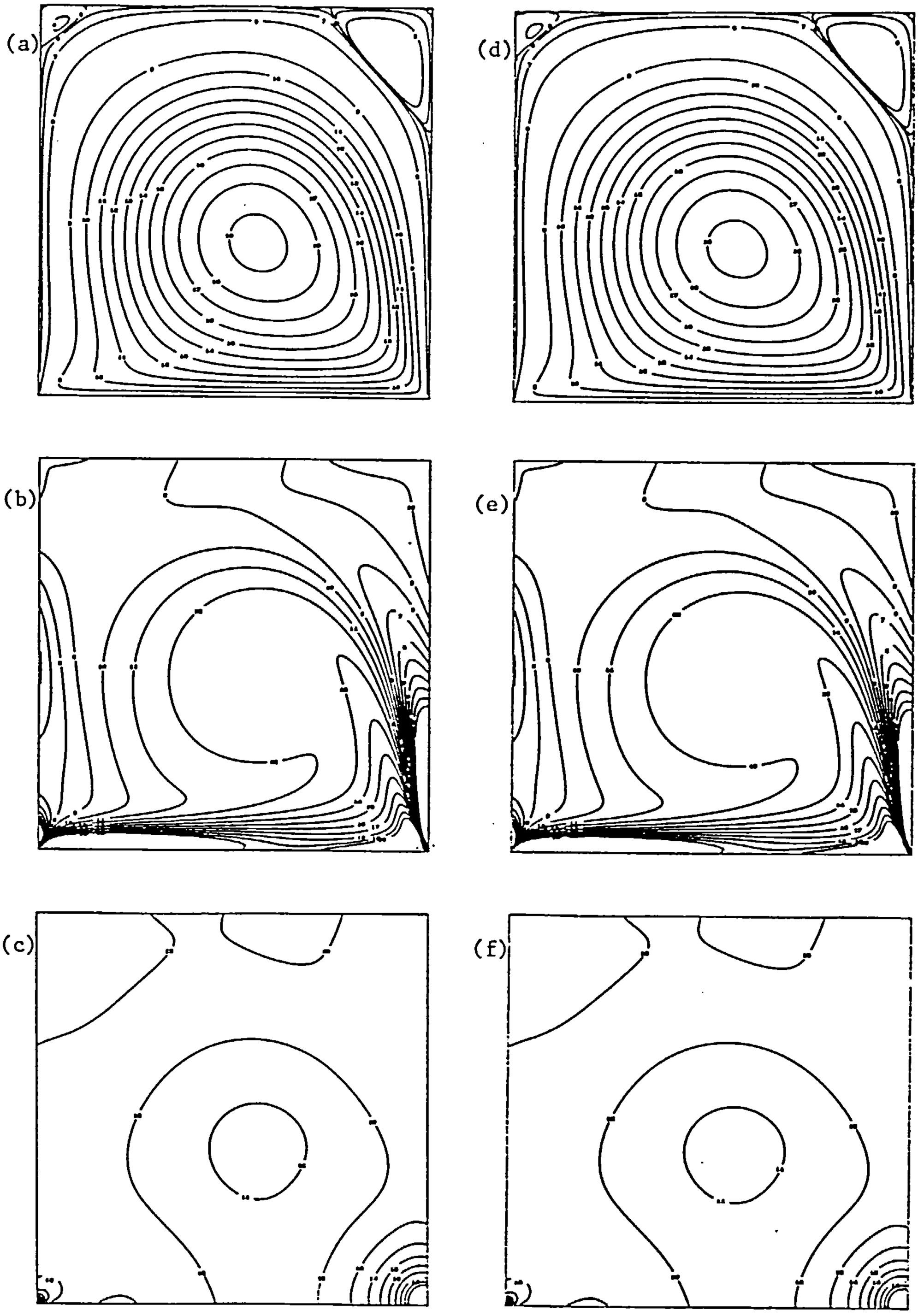


Figure 3.15: Contour plots for Reynolds number 400 with 128^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

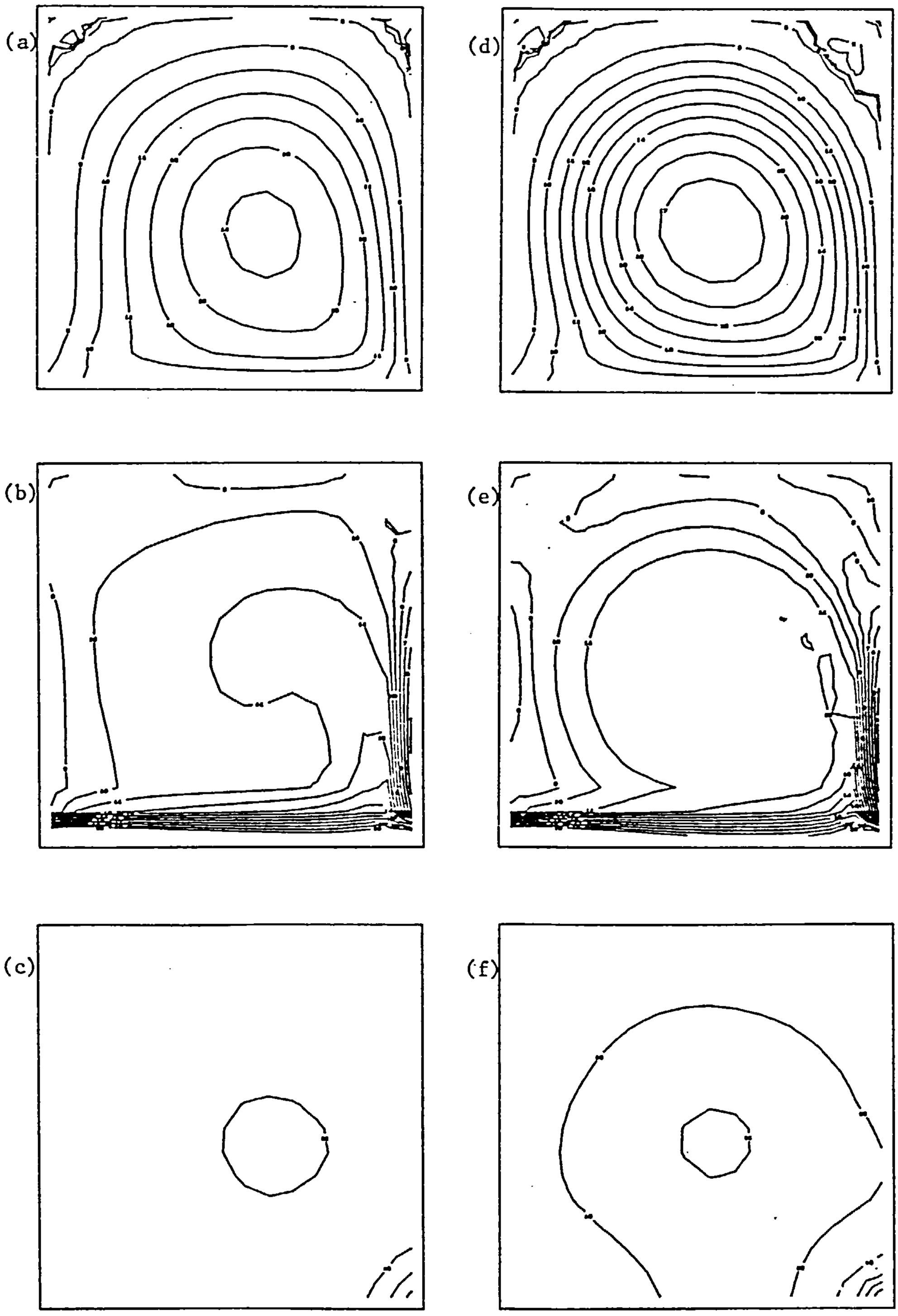


Figure 3.16: Contour plots for Reynolds number 1 000 with 16^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

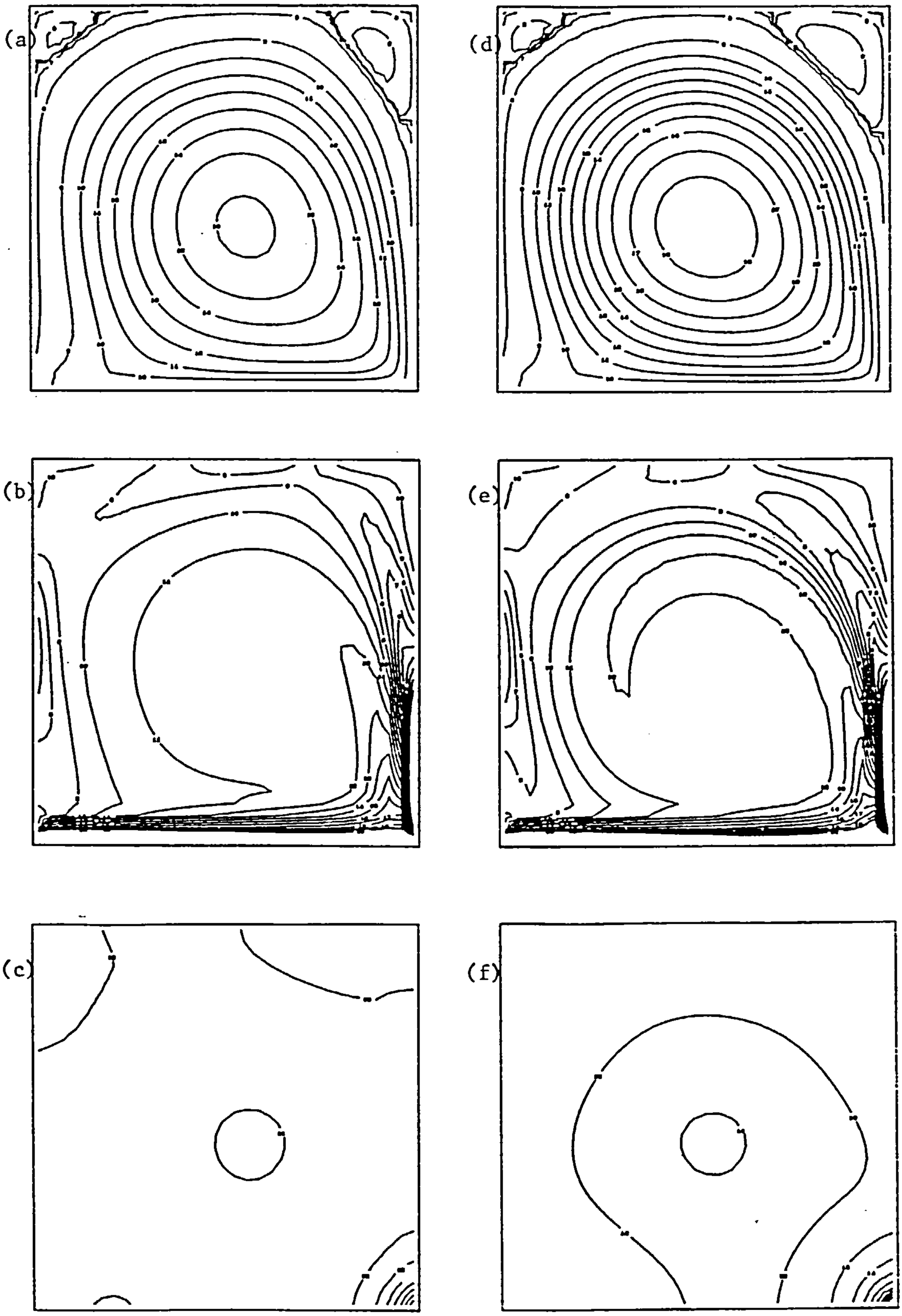


Figure 3.17: Contour plots for Reynolds number 1 000 with 32^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

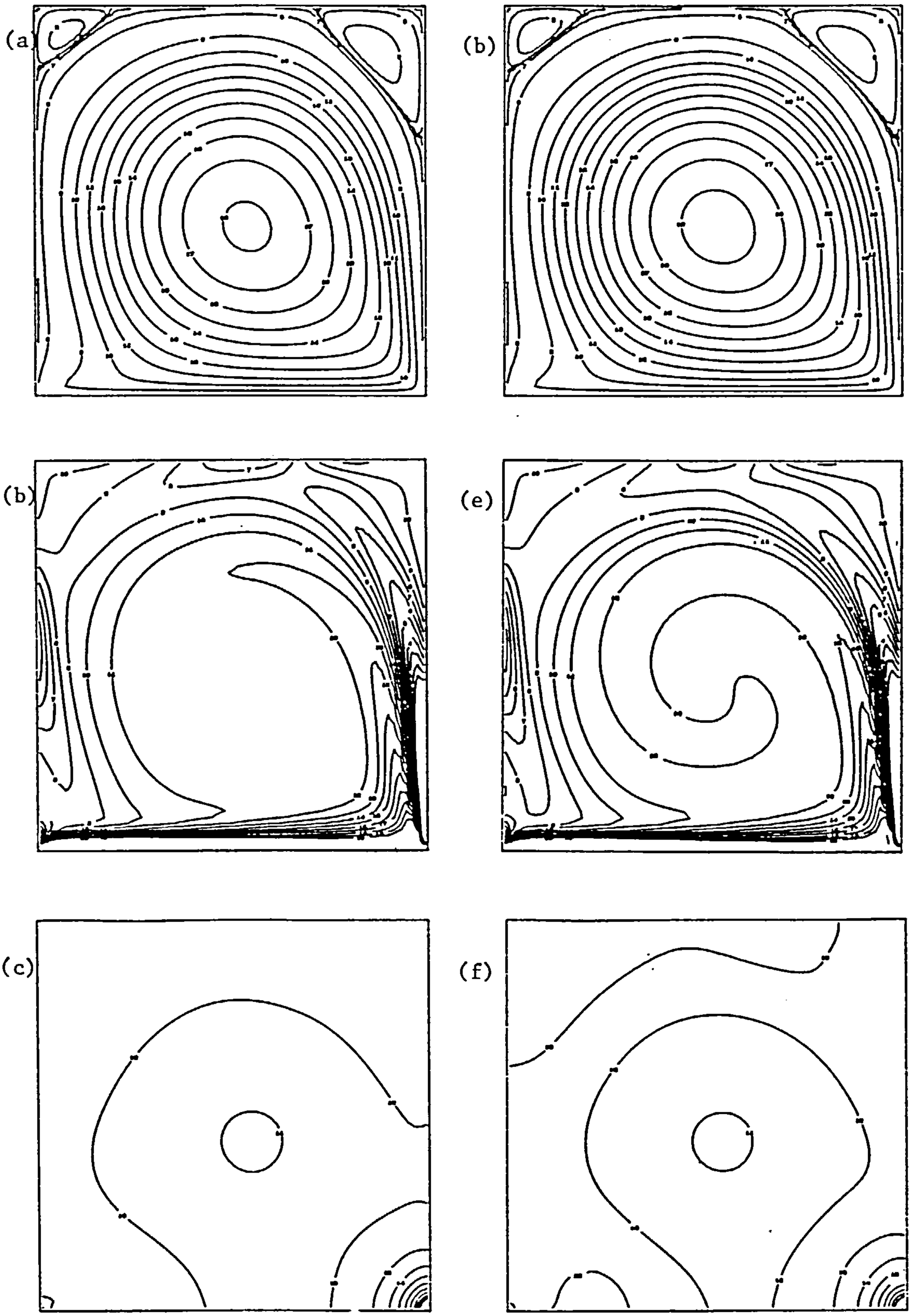


Figure 3.18: Contour plots for Reynolds number 1 000 with 64^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

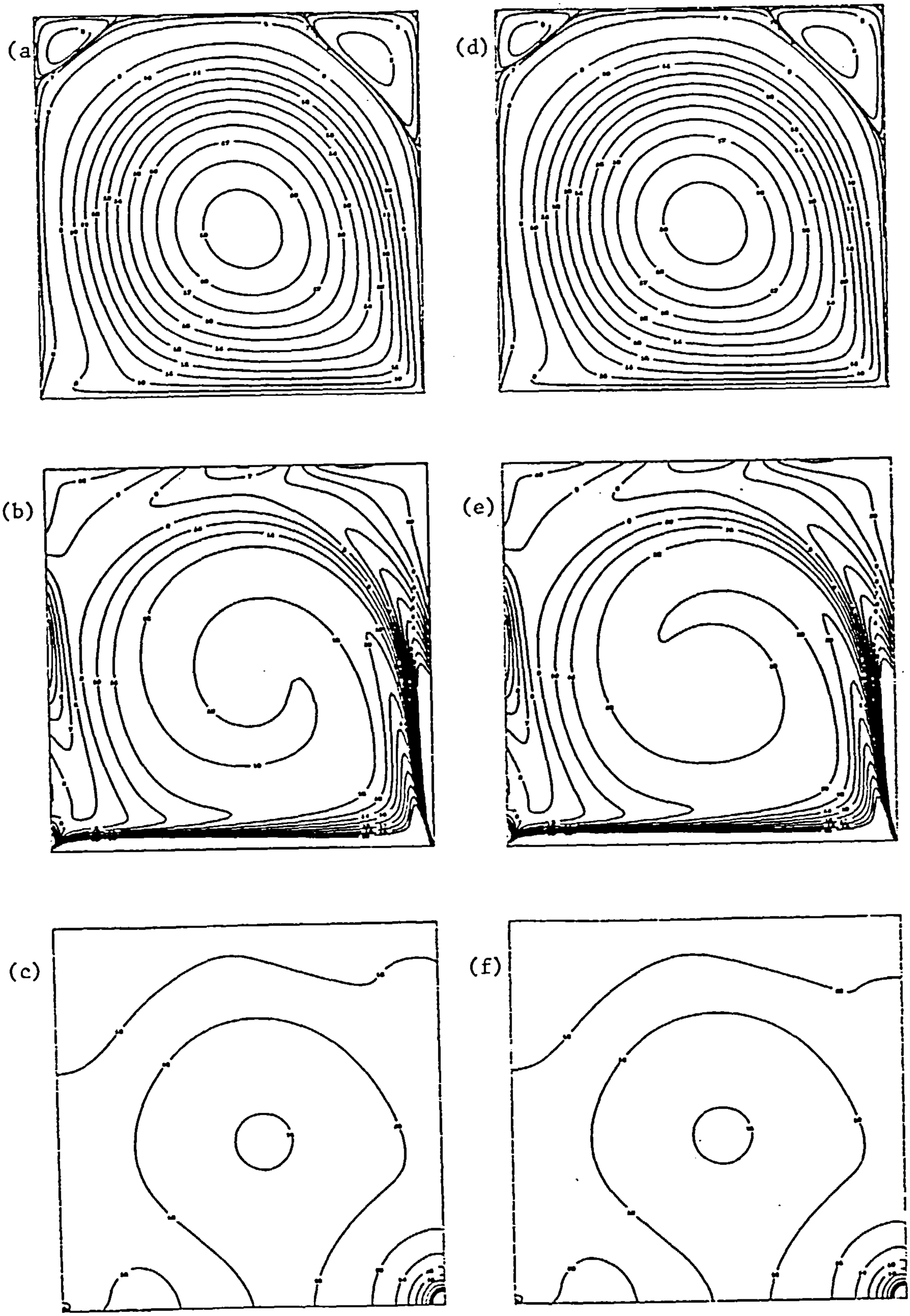


Figure 3.19: Contour plots for Reynolds number 1 000 with 128^2 internal nodes: hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

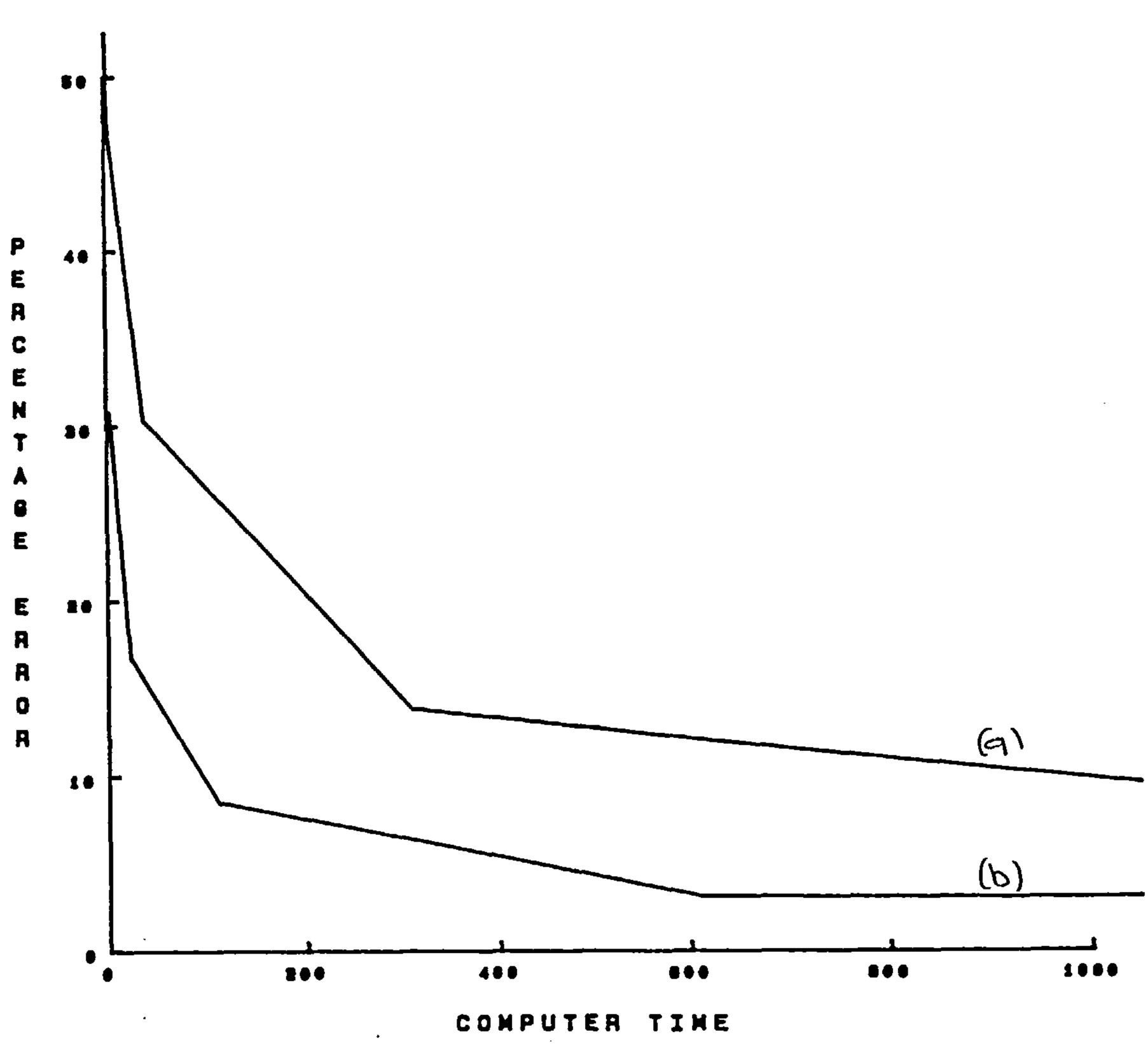


Figure 3.20: Plot of the percentage error against cpu time (seconds) obtained with (a) hybrid and (b) CCCT(α=0 for the lid-driven cavity problem.

References

1. Batchelor, G.K., *An Introduction to Fluid Dynamics*, CUP (1967).

2. Brandt, A. and Dinar, N., "Multigrid Solution to Elliptic Flow Problems", pp. 53-147 in *Numerical Methods in Partial Differential Equations*, ed. S. Parter, (1977).

3. Chleboun, P.V., "Mathematical Modelling Relevant to Gas Turbine Combustion", in *Ph.D. Thesis, University of Leeds*, (1987).

4. Doormaal, J.P. Van and Raithby, G.D., "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows", *Numerical Heat Transfer* **7** pp. 147-163 (1984).

5. Galpin, P.F., Doormaal, J.P. Van, and Raithby, G.D., "Solution of the Incompressible Mass and Momentum Equation by Application of a Coupled Equation Line Solver", *International Journal for Numerical Methods in Fluids* **5** pp. 615-625 (1985).

6. Galpin, P.F. and Raithby, G.D., "Numerical Solution of Problems in Incompressible Fluid Flow : Treatment of the Temperature-Velocity Coupling", *Numerical Heat Transfer* **10** pp. 105-129 (1986).

7. Gaskell, P.H. and Lau, A.K.C., "An Efficient Solution Strategy for Use with Higher Order Discretisation Schemes", *Report No. T40, Department of Mechanical Engineering, University of Leeds*, (1986).

8. Gaskell, P.H. and Lau, A.K.C., "Curvature Compensated Convective Transport : SMART, a New Boundedness Preserving Transport Algorithm", *International Journal for Numerical Methods in Fluids* **8** pp. 617-641 (1988).

9. Gaskell, P.H., Lau, A.K.C., and Wright, N.G., "Comparison of Two Strategies for Use with Higher Order Discretisation Schemes in Fluid Flow Simulation", *International Journal for Numerical Methods in Fluids - in press*, ().

10. Ghia, U., Ghia, K.N., and Shin, C.T., "High Re Solutions for Incompressible Flow Using the Navier- Stokes Equations and a Multigrid Method", *Journal of Computational Physics* 43(3)(December 1982).

11. Jacobs, D.H., "The Strongly Implicit Procedure for Biharmonic Problems", *Journal of Computational Physics* **13** pp. 303-315 (1973).

12. Jones, W.P., "Private Communication", *Imperial College of Science and Technology*, (1987).

13. Leonard, B.P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", *Computational Methods in Applied Mechanics and Engineering* **19** pp. 59-98 (1979).

14. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing (1980.).

15. Patankar, S.V., "A Calculation Procedure for Two-Dimensional Elliptic Situations", *Numerical Heat Transfer* **4** pp. 409-425 (1981).

16. Patankar, S.V. and Spalding, D.B., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows", *International Journal of Heat Mass Transfer* **15** pp. 1787-1806 (1972).

17. Raithby, G.D., "Skew Upstream Differencing Schemes for Problems Involving Fluid Flow", *Computer Methods in Applied Mechanics and Engineering* **9** pp. 153-164 (1976).

18. Raithby, G.D. and Schneider, G.E., "Numerical Solution of Problems in Incompressible Fluid Flow: Treatment of the Velocity-Pressure Coupling", *Numerical Heat Transfer* **2** pp. 417-440 (1979).

19. Shaw, G. and Sivaloganathan, S., "On the Smoothing Properties of the SIMPLE Pressure-Correction Algorithm", *International Journal for Numerical Methods in Fluids* **8** pp. 441-461 (1988).

20. Sivaloganathan, S. and Shaw, G.J., "An Efficient Non-linear Multigrid Procedure for the Incompressible Navier-Stokes Equations", pp. 221-233 in *Proceedings of the 5th International Conference on Numerical Methods in Laminar and Turbulent Flow, Part 1*, ed. Taylor, Habashi and Hafez, (1987).

21. Vanka, S.P., "Block-Implicit Calculation of Steady Turbulent Recirculating Flows", *International Journal of Heat Mass Transfer* **28**(11) pp. 2093-2103 (1985).

22. Vanka, S.P., "Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables", *Journal of Computational Physics* **65** pp. 138-158 (1986).

23. Zedan, M. and Schneider, G.E., "Investigation of Simultaneous Variable Solution Procedure for Velocity and Pressure in Incompressible Fluid Flow Problems", *AIAA-83-1519*, (June 1983).

24. Zedan, M. and Schneider, G.E., "A Strongly Implicit Simultaneous Variable Solution Procedure for Velocity and Pressure in Fluid Flow Problems", *AIAA-83-1569*, (June 1983).

Chapter 4

A MULTIGRID SOLUTION STRATEGY

4.1. Introduction

A serious constraint on the application of CFD techniques to real problems is the small number of nodes used. From the test problem considered in Chapter 3 we can see that a mesh of 64^2 nodes is required for accurate solutions, although one of 32^2 gives a qualitatively correct answer that would suffice in some situations. However, the solution of many problems has been attempted for very coarse meshes containing approximately 16^2 nodes, which are very inaccurate, especially when hybrid[19] differencing is used. In Chapter 3 the hybrid solution for the lid-driven cavity on a 16^2 mesh at a Reynolds number of 1 000 gives an error of 46% in the value of the maximum streamfunction.

When solution techniques are extended to three dimensions these problems are exacerbated. This has lead many researchers to deal with only two dimensional or axisymmetric problems, despite the fact that most flows of practical interest should be solved in three dimensions.

In the past, the main research activity aimed at overcoming the above problems has been the design of iterative schemes with higher error reductions. These still have the disadvantage of convergence rates tailing off, as described in section 3.3. The one idea that has opened the door to practical and accurate solution of such flows is the concept of multigrids, which was also outlined briefly in section 3.3. In this chapter the theory of multigrids is discussed more fully, various techniques are outlined and results obtained with the method adopted here are presented. The problem considered, being that of a two-dimensional lid-driven cavity. In subsequent chapters results are presented for more demanding flow situations.

4.2. Multigrid Theory

We now describe the theoretical basis of the multigrid algorithm in detail. The partial differential equations governing the flow (equation 3.1) are discretised to give the system

$$L^k Q^k = f^k, \tag{4.1}$$

where L^k is the discrete operator representing the partial differential operator, Q^k is the solution vector and f^k is the source term if any exists. The superscript k denotes the kth grid. Q^k contains all the variables to be found,

$$Q^k = (u_{11}v_{11}p_{11} \cdots u_{ij}v_{ij}p_{ij} \cdots u_{NN}v_{NN}p_{NN})^T.$$

At any stage of our iterative process we have an approximation to the exact solution Q^k: let us call this q^k. q^k does not satisfy equation (4.1) exactly, so there is a residual r^k

$$r^k = f^k - L^k(q^k). \tag{4.2}$$

We now define s^k, the difference between the approximation and the exact solution by

$$Q^k = q^k + s^k. \tag{4.3}$$

Then from equations (4.3) and (4.1)

$$L^k(q^k+s^k) = f^k, \tag{4.4}$$

and this gives, using equation (4.2),

$$L^k(q^k+s^k) = r^k + L^k(q^k). \tag{4.5}$$

If the operator L^k is linear we can now write

$$L^k(s^k) = r^k, \tag{4.6}$$

and solve this to obtain a correction. This method has been used very successfully for problems such as those governed by Laplace's equation. However the problems of interest to us are non-linear. We could linearise the operator L^k and solve equation (4.6) using a linear multigrid technique. An alternative method is preferred in which we solve for the variable q^k+s^k as a whole in equation (4.5). This is called Full Approximation Storage (FAS)[1]. With a multigrid approach we choose to solve equation (4.5) on the coarser grid k-1, where usually $h_{k-1}=2h_k$. Coarsening by a factor 2 is almost always employed. This simplifies interpolation (see sub-section 4.3.1), yet still gives a range of wavelengths close enough for efficient reduction of errors.

The new equation on grid k-1 is

$$L^{k-1}(I_k^{k-1}q^k+s^{k-1}) = I_k^{k-1}r^k+L^{k-1}(I_k^{k-1}q^k), \tag{4.7}$$

where I_k^{k-1} represents interpolation from grid k to grid k-1, called restriction. Relaxation is carried out on this grid (k-1) to a specified tolerance. Then s^{k-1} is calculated as the difference between the initial and final solutions on grid k-1. s^{k-1} is prolonged (interpolated) onto the grid k to give $I_{k-1}^k s^{k-1}$ which is added to q^k. In a similar way an equation on grid k-2 can be formed for equation (4.7) by restriction and this equation can then be restricted to give an equation on grid k-3, and so on. This will give a system of equations for grids k = 1, ...,m (m being the finest)

$$L^{k-1}(I_k^{k-1}q^k+s^{k-1}) = F^{k-1}, \tag{4.8}$$

where

$$F^{k-1} = I_k^{k-1}r^k+L^{k-1}(I_k^{k-1}q^k),$$

and

$$F^m=f^m.$$

It has been shown theoretically by Brandt and Dinar[1] (who first proposed these techniques) that such an algorithm will be very efficient. It should give convergence independent of grid size and so take the same number of iterations for all grids and γ (see sub-section 3.3) will be constant throughout.

4.3. Multigrid Techniques

The above theory has been extensively applied to linear problems, but only recently has it been applied to the problems of interest here. Some of the advances in this field are outlined below, in order to place the work presented here into context.

4.3.1. SIMPLE as a multigrid smoother

As mentioned in sub-section 3.4.1, Shaw and Sivaloganathan[18] have investigated the use of the SIMPLE[12] smoothing technique with an FAS algorithm. In their earlier

work[17] they showed that SIMPLE had reasonable theoretical smoothing rates. They have implemented the scheme with hybrid[19] differencing for the two-dimensional lid-driven cavity, using a staggered mesh arrangement, such that four fine grid continuity cells corresponded to one coarse-grid continuity cell. The algorithm they used proceeds as follows:

1) Pre-smoothing: The initial approximation is smoothed by applying ν_1 iterations of SIMPLE.

2) Coarse-grid correction: The coarse-grid problem is set up and solved; the correction is then transferred onto the fine grid.

3) Post-smoothing: The solution is smoothed by ν_2 iterations to eliminate any high frequency components introduced by the prolongation.

In practice a similar procedure is used to solve the equations at stage 2. This is done recursively until the coarsest grid is reached; here a . . solution is found to the prescribed accuracy.

The results obtained with this method showed h-independent convergence for grids containing up to 64^2 internal nodes. However, as Shaw and Sivaloganathan[18] point out in their paper, the accuracy of these results is restricted by the use of hybrid differencing.

4.3.2. SIMPLE with Linear Multigrid Techniques

Phillips and Schmidt[13, 14] have proposed an alternative means of using multigrids with SIMPLE. They use a linear multigrid technique[13] to solve the equations generated at each stage of the SIMPLE algorithm. So they use their technique to solve first the u-momentum equation, which has its coefficients frozen, and then similarly the v-momentum equation. If the problem involves a scalar quantity this is then solved for, again using a linear multigrid technique. Finally, a linear multigrid is used to solve the Poisson equation for pressure and the velocities are updated to ensure mass conservation.

This technique is not as efficient as the FAS algorithm because the solutions found at each stage are based on coefficients frozen at the previous one, whereas in FAS they are updated at every iteration. It is widely believed that obtaining very accurate solutions at each individual stage of SIMPLE is not beneficial in terms of the overall solution time. It is better to perform more outer iterations. In fact only very few iterations are required at each stage, and so the advantages of multigrid techniques are at best minimal.

Phillips and Schmidt implemented this multigrid technique with adaptive gridding and QUICK[8] discretisation. This gave accurate results for the lid driven and thermally driven cavity, for numbers of internal nodes upto 64. They did not present any cpu times or iteration counts, so it is not possible to make comparisons or to see if they found h-independent convergence.

This work was continued and analysed by Miller and Schmidt[11]. They use the SIMPLEC variant proposed by Raithby and van Doormal[2] and as before apply linear multigrids to the segregated equations. As in the earlier work of Phillips and Schmidt[14], QUICK is implemented in an unusual manner. Due to solution difficulties that they claim to encounter with QUICK, they use hybrid differencing in the discretisation and introduce QUICK via corrections in the source terms - the corrections which represent the difference between QUICK and hybrid differencing are evaluated from values at the previous iterate. This slows down convergence, especially as these values are 'frozen' as the algorithm passes over the grids and the segregated equations. Gaskell, Lau and Wright[5] have implemented QUICK with SIMPLE, and it has been employed here (see Chapter 3) for various problems without detriment to convergence. It is difficult to understand why Miller and Schmidt[11] found their approach necessary, when using QUICK directly would be much more efficient.

This technique was implemented for the two dimensional lid-driven cavity (described earlier) and a sudden contraction (having fixed rather than derivative boundary conditions). The mesh consisted of 32^2 nodes, which is fairly coarse. For the sudden expansion, reductions in cpu time, upto a factor of five times, were achieved.

However, for the driven cavity the largest reduction was 37 % and in some cases the multigrid technique required more time. h-independence was never completely achieved and the cpu times are greater by a factor of about 10 over those reported here.

4.3.3. DGS as a multigrid smoother

In their paper Brandt and Dinar[1] presented results for an FAS algorithm coupled with DGS smoothing. The multigrid strategy was an adaptive one and the smoothing process was transferred to a coarser grid when the convergence rate increased beyond a specified value. The solution was sought on the coarser grid until the initial residual had been reduced by a specified factor and then the correction was transferred to the fine grid. The coarse-grid problem could in turn be transferred to a yet coarser grid, etcetera.

This was applied to an idealised problem with good results. However, it has not been extended to more complex flows. In fact, Linden Stecken and Stuben[9] show that for higher Reynolds numbers the technique will not converge at all.

4.3.4. A Multigrid CELS approach

Hutchinson, Galpin and Raithby[7] outline the solution of the Navier-Stokes equation using a technique called Additive Correction[16] which is conceptually similar to multigridding. They applied this to the smoother CELS, outlined in Chapter 3, and with Raithby's Upstream Weighted Difference Scheme[15] (UWDS). With Additive Correction (AC) the coarse-grid equations are derived directly from interpolation of the fine grid equations. However, it is the linearised fine grid equations that are transferred, so it would appear that this technique corresponds to the usual linear multigrid approach for the fine grid equations. This linear multigrid is used to find the solution at each time step of a false transient method.

Hutchinson et al solved the two-dimensional driven cavity problem for grids having up to 96^2 internal nodes. Solutions times were reduced by a factor of 8.7 over non-AC CELS. It is difficult to say from the results presented whether h-independent

convergence was achieved. A thermal problem was also solved resulting in a reduction factor of approximately 2.

4.3.5. Segregated or Unsegregated Smoothing Techniques?

The techniques outlined in sub-sections 4.2.1-4.2.3 are all segregated (see 3.4.2); they solve for u, v and p separately. The CELS approach outlined in 3.4.3 is unsegregated because it solves for u, v and p simultaneously on a line. Unsegregated methods are better at smoothing because they take into account all errors simultaneously and so introduce fewer spurious modes. This has been borne out by Linden et al[9], who state that "*compared to the above* unsegregated *multigrid approaches* segregated multigrid approaches are *relatively expensive*". Techniques that only use multigrid for the linearised problems are even less advantageous, for similar reasons.

4.4. Multigrid with Block Implicit Method

In Chapter 3 a Block Implicit smoothing technique was presented. This technique is simple to implement, and stable. It has full unsegregated coupling of momentum and continuity and has been successfully applied to the driven cavity problem. In view of this it was decided to use it here in the evaluation of a multigrid approach.

Several multigrid strategies have been proposed and implemented for cycling between grids in order to eliminate all errors efficiently. The one used here differs from those of Shaw and Sivaloganathan[18] and Brandt and Dinar[1] and is outlined below.

(i) The system of equations in (4.8) is set up on all grids k = 1, ..., m.

(ii) The residual of the system on each level is calculated. The one with largest residual is selected for smoothing.

(iii) The solution on this grid is smoothed using the Block Implicit Method. This is done until the residual has been reduced by a factor η.

(iv) The difference between the initial and final solution on the present grid is calculated. This is the correction for the next finer grid, so it is prolonged and added to the solution there.

(v) If the new grid is not the finest, M, we repeat (iii) and (iv).

(vi) If we have reached the finest grid, it is smoothed until the error has been reduced by a factor η.

One multigrid iteration consists of executing the above sequence once. Iterations are continued until the residual on the fine grid is less than a specified tolerance.

This multigrid strategy is based on the idea that the level with the largest residual is the one on which it is most efficient to work at that stage. When this error has been eliminated, another grid may be selected. This process is continued until all errors are eliminated and the fine grid problem has converged. This strategy was first suggested by Falle and Wilson[3] and has been successfully implemented by them for a variety of problems.

The residual measure and tolerance are the same as were used for the ordinary BIM described in Chapter 3. It can be seen that the formulation of the BIM in terms of residuals and updates (as described in 3.4.6) is advantageous for use with multigrids, as the values of the residuals are readily available for transfer to coarser grids.

4.4.1. Restriction and Prolongation

The grid coarsening adopted here is outlined in Figure 4.1. As stated earlier a mesh reduction factor of 2 is used. One coarse-grid scalar cell coincides with four fine grid scalar cells. The velocity cells for each grid are orientated in line with what one would expect with a procedure for a one-grid technique; this means that coarse and fine grid velocity cells do not coincide.

The nature of the restriction and prolongation operators is dictated by the mesh arrangement. Linear interpolation is used for the restriction of velocities, and bi-linear interpolation for the restriction of scalars. The boundaries are not restricted here as

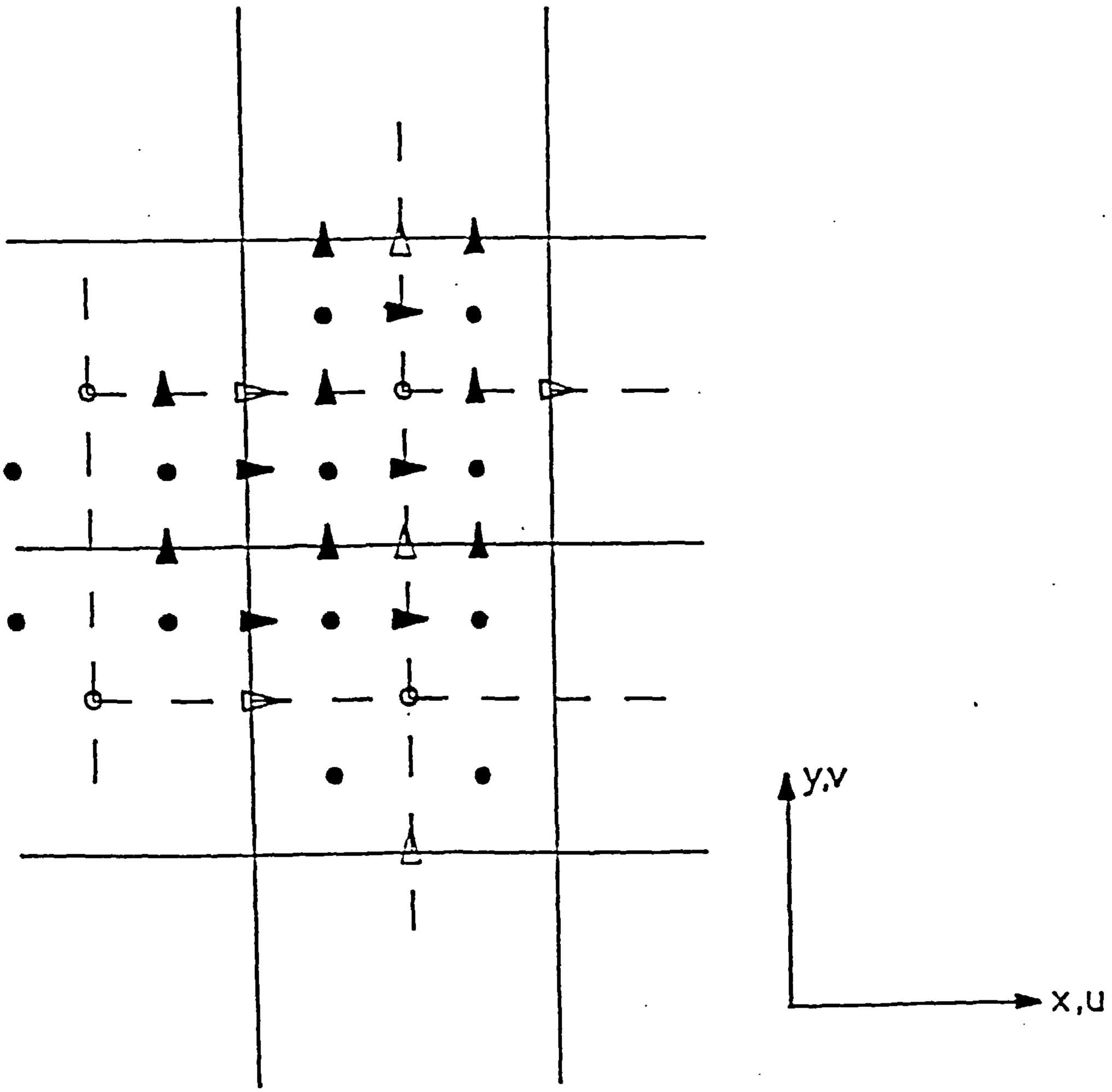


Figure 4.1: Multigrid coarse to find nodal configuration: ▷ coarse grid velocity, ▶ fine grid velocity, O coarse grid scalar, ● fine grid scalar.

they are set at fixed values. Prolongation is also carried out by linear and bi-linear interpolation. The exact form of the equations for both restriction and prolongation is given in the Appendix I. The use of quadratic interpolation is reported later.

However, one aspect of coarsening and interpolation should be mentioned here. Consider the right hand side of equation (4.7),

$$I_k^{k-1} r^k + L^{k-1}(I_k^{k-1} q^k). \tag{4.9}$$

Now, consider the point (i_c,j_c) on the coarse-grid (see Figure 4.2) and the four associated fine grid points $(i_f,j_f),(i_f+1,j_f),(i_f,j_f+1)$ and (i_f+1,j_f+1). At this point the continuity equation is represented by

$$I_k^{k-1} r^k = \frac{-1}{4}\frac{u_{i_c j_c} - u_{i_c-1\, j_c}}{h} + \frac{v_{i_c j_c} - v_{i_c\, jf-1}}{h} \tag{4.10}$$

$$+ \frac{u_{if+1\, j_c} - u_{i_c j_c}}{h} + \frac{v_{if+1\, j_c} - v_{if+1\, jf-1}}{h}$$

$$+ \frac{u_{i_c\, jf+1} - u_{if-1\, jf+1}}{h} + \frac{v_{i_c\, jf+1} - v_{i_c j_c}}{h}$$

$$+ \frac{u_{if+1\, jf+1} - u_{i_c\, jf+1}}{h} + \frac{v_{if+1\, jf+1} - v_{if+1\, j_c}}{h}$$

$$= \frac{-1}{4h}\left[(u_{if+1\, j_c} + u_{if+1\, jf+1}) - (u_{if-1\, j_c} + u_{if-1\, jf+1}) + (v_{i_c\, jf+1} + v_{if+1\, jf+1}) - (v_{i_c\, jf-1} + v_{if+1\, jf-1})\right],$$

and

$$L^{k-1}(I_k^{k-1} q^k) = \frac{u_{i_c j_c} - u_{ic-1\, j_c}}{2h} + \frac{v_{i_c j_c} - v_{i_c\, jc-1}}{2h} \tag{4.11}$$

$$= \frac{1}{4h}\left[(u_{if+1\, j_c} + u_{if+1\, jf+1}) - (u_{if-1\, j_c} + u_{if-1\, jf+1}) + (v_{i_c\, jf+1} - v_{if+1\, jf+1}) + (v_{i_c\, jf-1} + v_{if+1\, jf-1})\right].$$

So that

$$I_k^{k-1} r^k + L^{k-1}(I_k^{k-1} q^k) = 0.$$

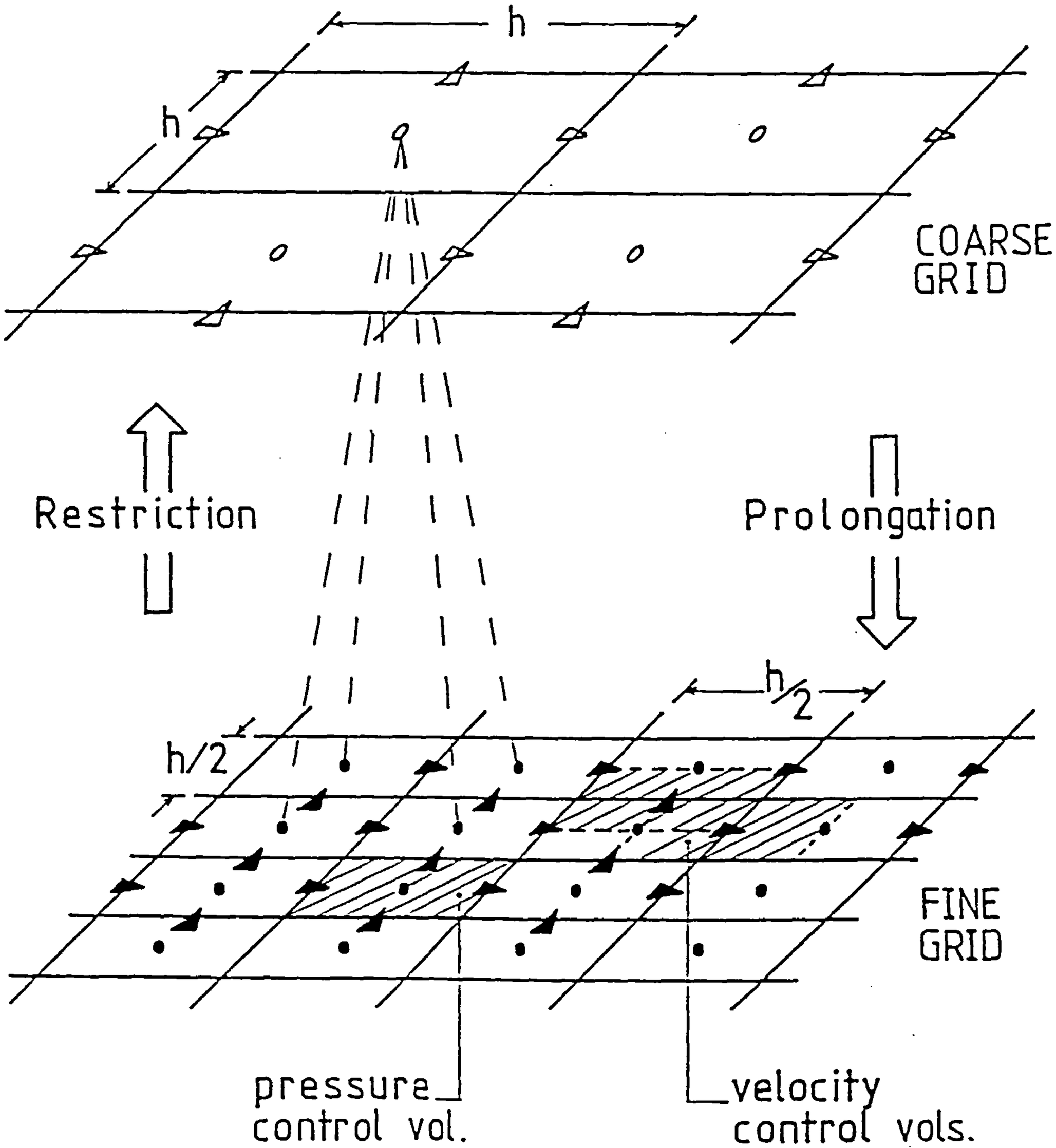


Figure 4.2: Illustration of coarse to fine restriction of the continuity equation.

Thus the coarse-grid right hand side for the continuity equation is zero. When implementing the algorithm this right hand side could be computed along with that for the other equations. However, this is seen to be unnecessary from the above analysis. In fact it would be disadvantageous to do so, as this would give values of 'round off' error rather than zero.

Sivaloganathan and Shaw[18] have also discussed this aspect. They refer to it as "continuity satisfaction", i.e. if continuity is satisfied on the fine grid it is automatically satisfied on the coarse-grid. They also state that this method of coarsening gives rise naturally to compatible momentum control volumes.

Lonsdale[10] used a different configuration, based on a non-uniform grid. The pressure lines of the finest grid are positioned at points corresponding to the zero of the relevant Chebyshev polynomial, so as to cluster points near the boundary. As in our approach, velocity lines are positioned halfway between pressure lines. Coarsening is done by taking the first and every other fine grid pressure line as a coarse-grid pressure line. This maintains the grid line next to the boundary, and is advantageous with respect to the handling of coarse-grid boundary conditions. Overall this technique has the benefit of fine grid resolution near-boundaries, but the grid positions and coarsening makes it more complex and less adaptable to different geometries and for use with higher order discretisation schemes.

During prolongation, special attention has to be paid to the near-boundary points because of the grid configuration at the boundaries. The equation for a near-boundary cell is different from the others. If the boundary values are used to calculate the near-boundary values it is found that prolongation can make the method unstable at high Reynolds numbers. It is better to use a zero derivative condition for these points and put them equal to their value on the first coarse-grid line. This is not as accurate as one might hope, so after each prolongation all the boundary adjacent cells are updated once. It has been observed that these measures can reduce cpu time by up to 30%, compared with prolongation using boundary values alone (see Table 4.1), and, as mentioned earlier, ensure stability.

Re	100		1000	
	b	nb	b	nb
4	0.11	0.11	0.24	0.24
8	0.63	0.63	2.77	3.13
16	1.97	2.04	7.44	11.53
32	5.72	5.86	23.65	35.03
64	19.71	22.53	75.01	115.52
128	77.01	75.89	228.05	332.80
256	303.03	335.64	848.69	1026.36

Table 4.1: Computer time for Reynolds numbers of 100 and 1 000 with CCCT(α=0) differencing with (b) and without (nb) boundary iterations.

4.5. Application of the Multigrid Algorithm with a Block Implicit Method

The above method was applied to the test problem described in Chapter 3, namely the two-dimensional driven cavity (see Figure 3.1). Solutions were obtained for Reynolds numbers of 100, 400 and 1 000 with hybrid and CCCT(α=0)^4 discretisation. The largest number of internal nodes was 256^2. A Full Multigrid (FMG) algorithm was used, i.e. a solution on a coarse-grid was obtained and prolonged to form an initial solution on the next fine grid. On the coarsest grid initial conditions of zero were used. i.e. u=v=p=0 for all x,y.

4.5.1. Computational Details

The relaxation factors adopted were those reported in Chapter 3 for the Block Implicit Method. It may be advantageous to vary these, but the aim here is to demonstrate robustness and so fine tuning was not carried out. With respect to the pressure condition, p(0.5,0.5) = 0.0, Brandt and Dinar[1] says that this should only be applied on one grid. It was found, however, that applying it on all grids had no effect on the convergence rate. It was sufficient, though, to only apply it on the coarse-grid. This strategy was adopted to save cpu time. Sivaloganathan and Shaw[18] have also observed this phenomenon.

The residuals that are transferred to the coarse-grid can be calculated "dynamically". That is, at each point the residual, calculated according to equation (3.31), can be stored in an array ready to be restricted to the coarse-grid at a later stage.

This residual is based on values that have yet to be updated and therefore is not the same as the actual residual for the solution after all updates have been made. An alternative approach is to calculate the correct residuals with an extra sweep, in which no updating is done and only residuals are calculated. Such a sweep obviously takes time, but only a fraction of that required for one update sweep. Although the latter method takes more time than one employing dynamic residuals, it is more than compensated for by better error correction and stability. When using dynamic residuals to set up a coarse-grid problem the set of equations often do not have a solution and cannot be solved to the order of round-off errors.

Some attention must be paid to the choice of the factor η in the algorithm. Different values of η can be used on different grids. Here η is restricted to three values. One is for the finest grid, one for the coarsest grid and one for all the intermediate grids. In earlier work[6] these were taken to be 0.1, 0.1 & 0.5 for the coarsest, intermediate and finest grids respectively. They were chosen so as to ensure that most of the smoothing was carried out on the coarse-grids where η=0.1. On the fine grid we wish only to eliminate any errors introduced by the interpolation, and so take η=0.5. These factors generated solutions that exhibited h-independent convergence behaviour, and were orders of magnitude faster than the ordinary Block Implicit Method.

As described earlier, when a fine grid problem is transferred to a coarse-grid, this new problem can then be transferred to an even coarser grid for solution, and so on. Eventually the solution is sought on the coarsest grid. Brandt and Dinar[1] suggest solving this problem exactly. An alternative technique can be used to find a solution on this grid (e.g. Newton's method), because of the small number of points.

In view of these ideas it was decided to investigate the use of a factor η=0.001 on the coarsest grid. This would give an accurate solution to the coarsest grid problem, without spending excessive amounts of time converging to round-off. The latter was not thought desirable for a problem that was only there to find an intermediate solution for the non-linear problem.

This strategy was found to improve convergence (see Table 4.2). Not only are the solution times less, but the multigrid convergence factor θ (see equation (4.16)) is less. In view of the large savings in cpu time these factors were selected for use with all other cases.

4.5.2. Results

The results for Reynolds numbers 100, 400 and 1 000, and grids of 4^2 upto 256^2 internal nodes, with hybrid and CCCT are presented in Table 4.3. For each case the total cpu time and the number of fine grid work units (FGWU) required are given. The fine grid work unit is a measure of the total work done on all grids expressed in terms of the work required for one fine grid iteration. For example, if the fine mesh is 128^2, four iterations on a mesh of 64^2 is one FGWU, and 64 iterations on a mesh of 16^2 is one FGWU.

At a first glance at the results, it can easily be seen that the times required are significantly less than those for an ordinary block implicit method (see Table 4.4 - the bracketed numbers are predictions based on the power law relation given in section 3.2). On the finest grid the cpu times for the multigrid solution are around 1% of those for the non-multigrid case. On coarse grids savings are significant although not as dramatic. Multigrids are most optimal for a large range of grids, that cover all the different wavelengths of error. Coarser grids cannot cover the same range as finer

Re	100		1000	
grid	(a)	(b)	(a)	(b)
4^2	0.11	0.11	0.24	0.24
8^2	0.45	0.63	3.83	2.77
16^2	2.04	1.97	9.95	7.44
32^2	6.62	5.72	37.35	23.65
64^2	21.17	20.33	105.58	75.01
128^2	89.10	77.01	372.09	228.05
256^2	332.55	303.03	1923.32	848.69

Table 4.2: Comparison of computer times (secs.) with CCCT(α=0) for different sets of values for the factor η, (a) (0.1,0.1,0.5) (b) (0.001,0.5,0.5)

a) Reynolds number = 100

	hybrid		CCCT	
	Cpu(secs.)	FGWU	Cpu(secs.)	FGWU
4	0.12	7.00	0.11	5.00
8	0.55	13.25	0.63	14.25
16	1.82	11.75	1.97	11.75
32	6.36	10.48	5.72	8.56
64	19.71	8.13	20.33	7.63
128	71.62	7.38	77.01	7.26
256	277.85	7.18	303.03	7.15

b) Reynolds number = 400

	hybrid		CCCT	
	Cpu(secs.)	FGWU	Cpu(secs.)	FGWU
4	0.15	10.00	0.14	8.00
8	0.76	19.25	1.20	30.00
16	2.84	18.88	4.03	25.63
32	10.01	16.91	12.87	20.38
64	35.55	15.13	36.23	14.12
128	104.25	10.96	94.99	9.08
256	311.04	8.09	321.64	7.63

c) Reynolds number = 1000

	hybrid		CCCT	
	Cpu(secs.)	FGWU	Cpu(secs.)	FGWU
4	0.12	7.00	0.24	20.00
8	0.84	21.50	2.77	74.00
16	2.38	15.69	7.44	49.00
32	12.47	21.52	23.65	38.5
64	65.88	29.16	75.01	30.19
128	213.95	23.36	228.05	22.57
256	474.87	12.66	848.69	20.94

Table 4.3: Solution time and FGWU for multigrid solutions at Reynolds numbers of 100, 400 and 1 000.

Re	100		400		1000	
	NMG	MG	NMG	MG	NMG	MG
4	0.11	0.11	0.14	0.14	0.24	0.24
8	0.35	0.63	0.85	1.20	3.20	2.77
16	2.33	1.97	5.71	4.03	23.37	7.44
32	19.40	5.72	37.29	12.87	112.15	23.65
64	264.23	20.33	279.59	36.23	611.12	75.01
128	3541.88	77.01	3772.66	94.99	7078.15	228.05
256	(29670.73)	303.03	(12828.00)	321.64	(66811.06)	848.69

Table 4.4: Computer time (secs.) for multigrid (MG) and non-multigrid (NMG) methods with CCCT(α=0) differencing.

ones and so do not give such large reductions in cpu time. It is not certain that ordinary solutions could be found on meshes with 256^2 points - such meshes may make ordinary schemes unstable. The only reason that solutions were not obtained on still finer grids was the limited amount of memory space available on our computer. This could have been overcome by writing values in parts of the solution domain onto disk and recalling them when necessary. This would however have increased the computing time greatly, and therefore it was decided not to proceed in this way, but to wait for a bigger computer. It should also be noted that when compared with earlier work[6] these computer times are found to be much less, due to the improved multigrid behaviour and the use of four sweeps in one iteration.

It is interesting to analyse Table 4.3 in detail. Cpu times increase between each grid by a factor less than four. This is borne out by inspecting the FGWU figures. One FGWU on a mesh of size h is equivalent to four FGWU on a mesh of size 2h, so if the power law relationship is obeyed the FGWU counts on each grid will be identical. In fact it can be seen that they decrease and so the relation is more than satisfied. This is explained by the optimality of fine grids detailed in the last paragraph. The relationship $\text{cpu} \propto N$ is equivalent to the property of h-independence, so we have achieved h-independence here.

Consider the definition of γ,

$$\gamma = \frac{R^n}{R^{n-1}}, \tag{4.12}$$

where R^n is the residual at stage n. Then

$$R^n = \gamma R^{n-1}, \tag{4.13}$$

and so if γ is constant

$$R^n = \gamma^n R^0, \tag{4.14}$$

and

$$\log R^n = n\log\gamma + \log R^0. \tag{4.15}$$

Therefore, a graph of log R against n, or in this case FGWU, should result in a straight line with negative gradient ($\gamma < 1 \Rightarrow \log\gamma < 0$). If γ increases then the gradient of this graph will increase. In view of this, Figures 4.3-4.8 show log R plotted against FGWU. For coarser grids it can be seen that the gradient increases and therefore the rate of reduction of the residual is increasing. On these grids multigridding is not optimal, because too few grids are being used. As grids get finer and more grids are used, the gradient becomes constant and so multigridding is operating optimally.

Using Table 4.3 one can compare the two discretisations. First, it should be noted that one CCCT FGWU requires up to 10% more cpu time than one hybrid FGWU, depending on the mesh size. Overall CCCT requires more work units than hybrid, though this is only significant for the 256^2 mesh solution at a Reynolds number of 1 000. In some cases CCCT is actually quicker than hybrid. The greater cpu requirement at Reynolds number 1 000 is explained by the crude linearisation employed in adapting CCCT to the Block Implicit Method (see sub-section 3.4.6). The non-linearities become more dominant as the Reynolds number increases. In Figures 4.9 to 4.11, the solutions for CCCT and hybrid are indistinguishable on a mesh of 256^2 internal nodes, but Table 4.5 shows values of key variables, for which there is a noticeable difference. Table 4.6 shows these key variables for the results of Ghia, Ghia and Shin,which are in very good agreement with those obtained here. As seen in

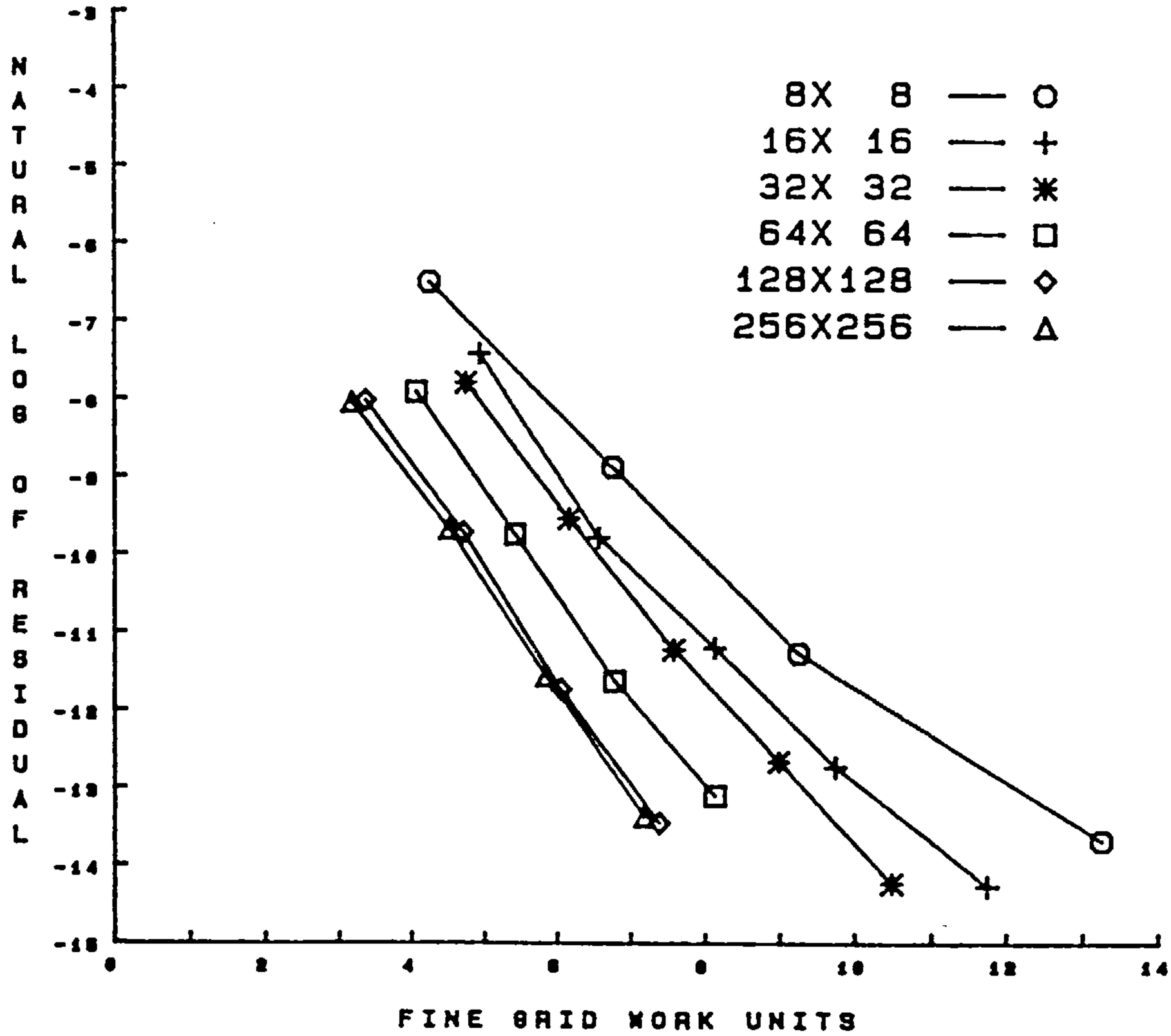


Figure 4.3: Plot of natural log of the residual against FGWU, for Reynolds number 100 and hybrid differencing.

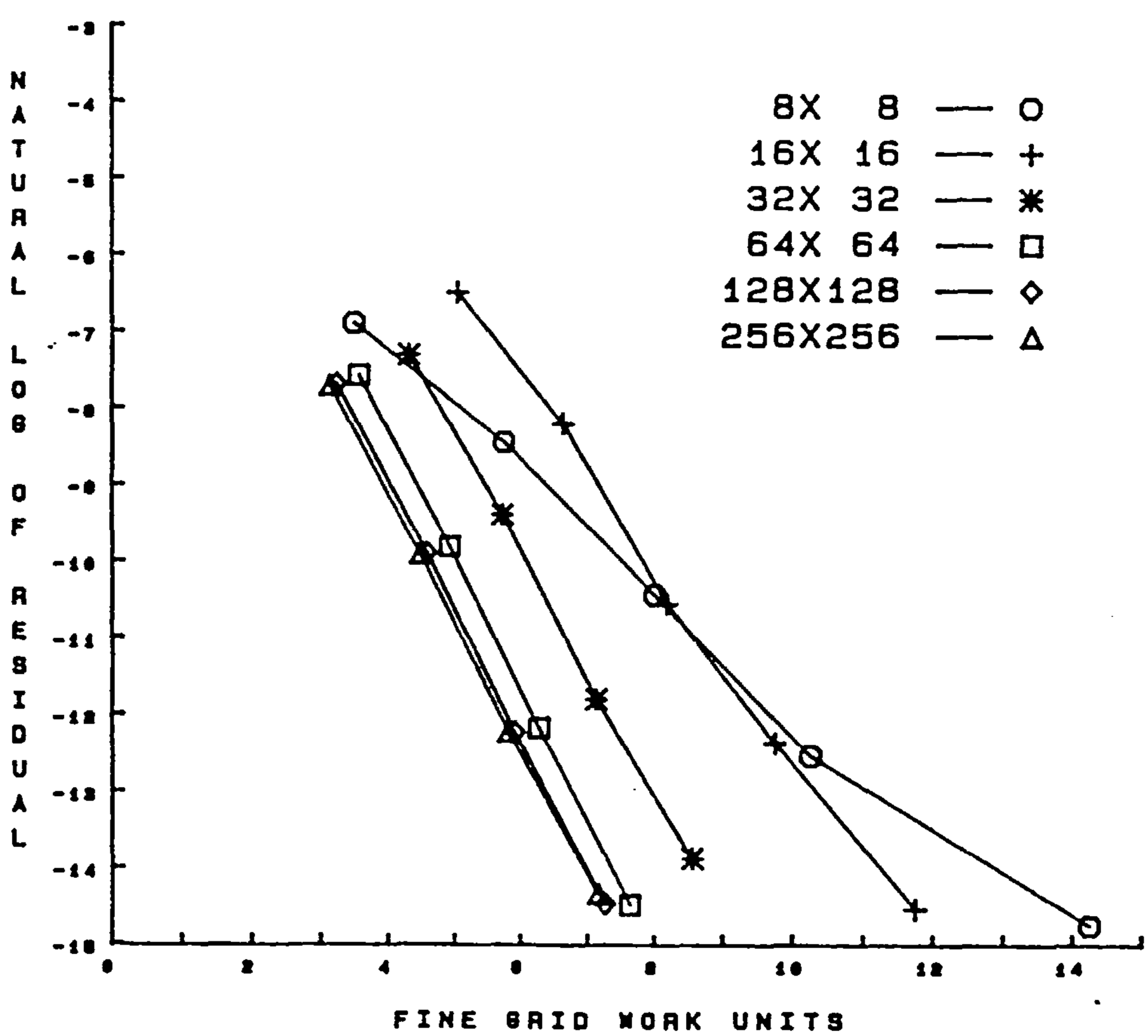


Figure 4.4: Plot of natural log of the residual against FGWU, for Reynolds number 100 and CCCT ($\alpha = 0$) differencing.

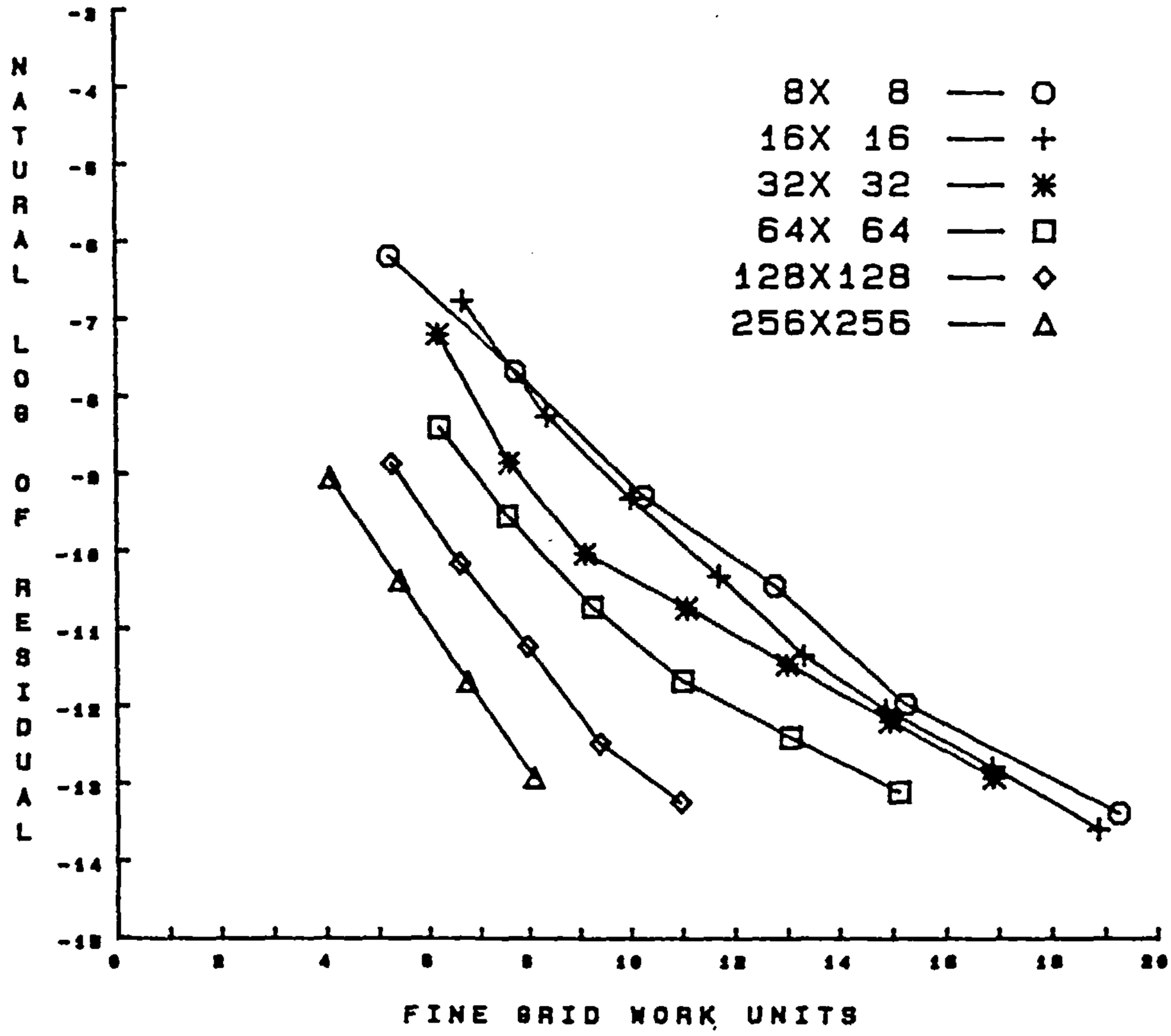


Figure 4.5: Plot of natural log of the residual against FGWU, for Reynolds number 400 and hybrid differencing.

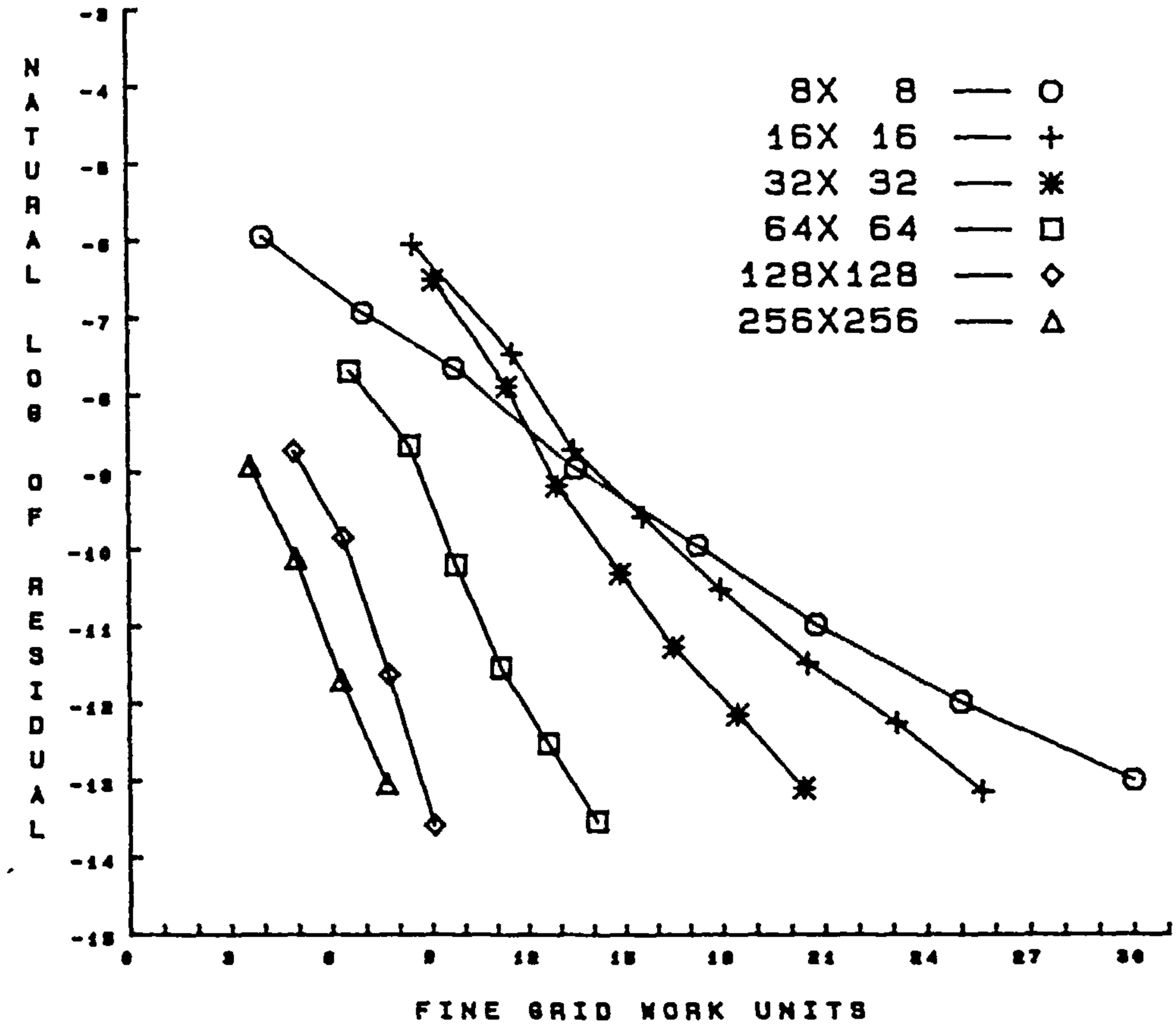


Figure 4.6: Plot of natural log of the residual against FGWU, for Reynolds number 400 and CCCT ($\alpha = 0$) differencing.

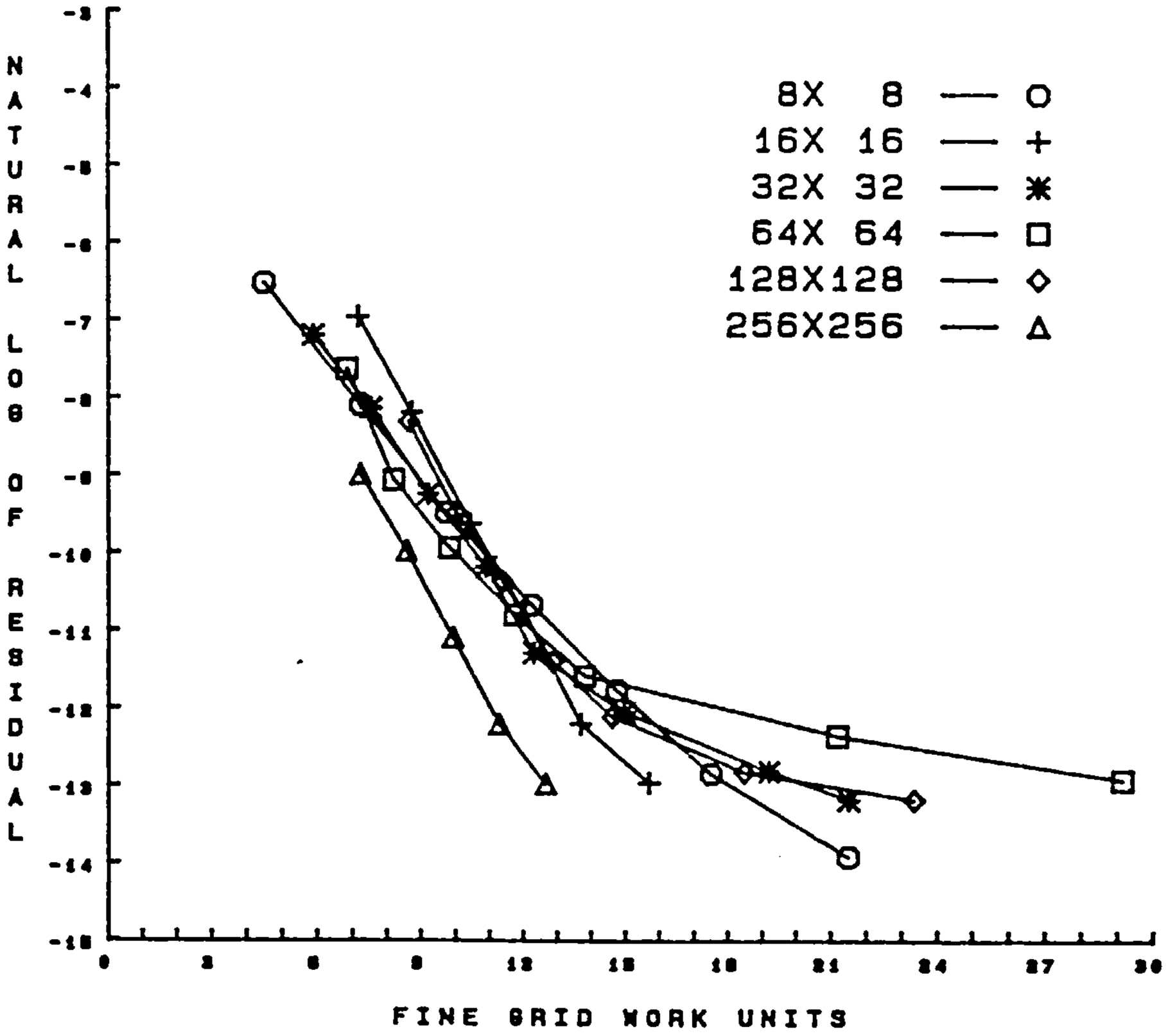


Figure 4.7: Plot of natural log of the residual against FGWU, for Reynolds number 1 000 and hybrid differencing.

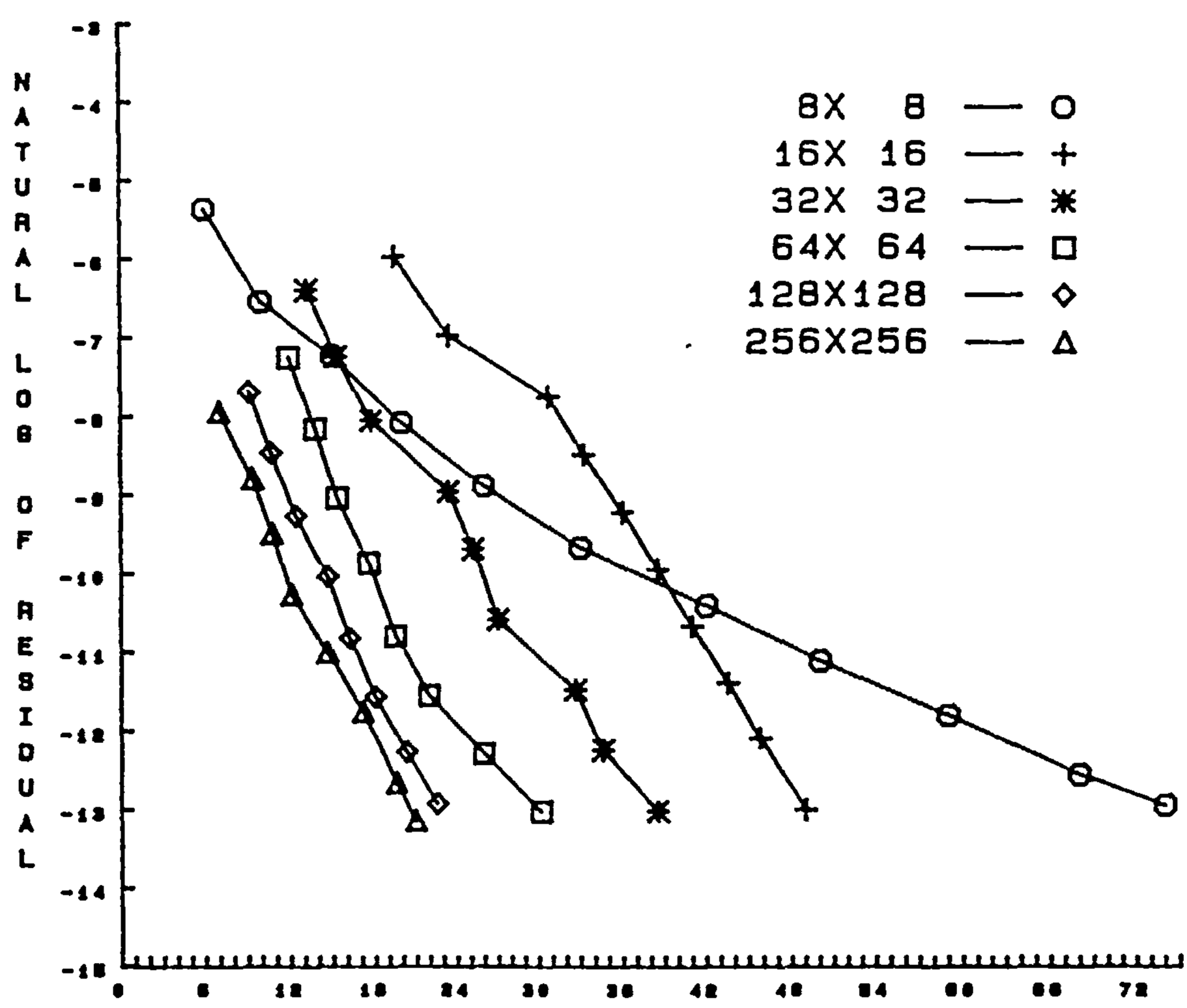


Figure 4.8: Plot of natural log of the residual against FGWU, for Reynolds number 1 000 and CCCT ($\alpha = 0$) differencing.

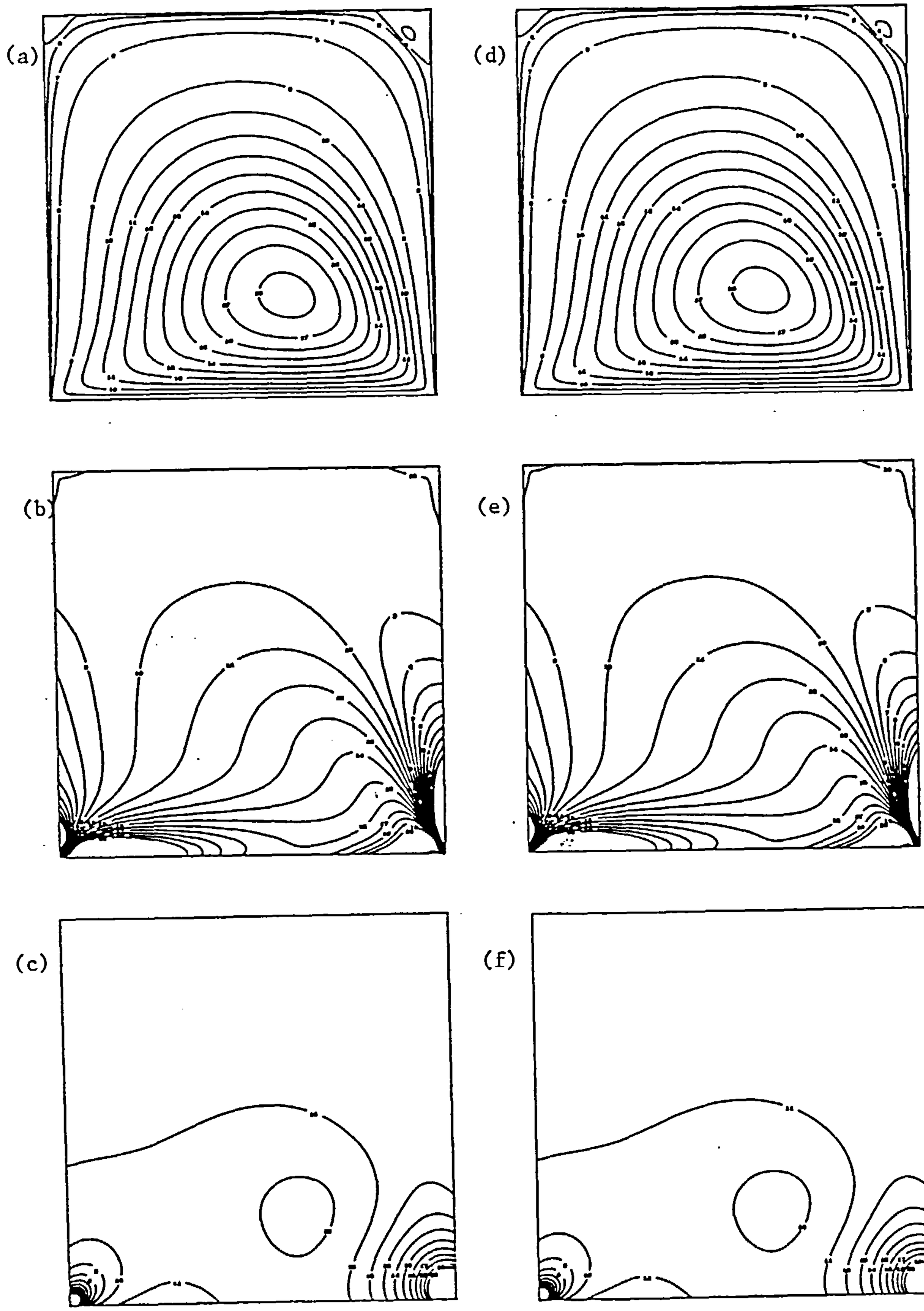


Figure 4.9: Contour plots for Reynolds number 100 with 256^2 internal nodes; hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e), streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

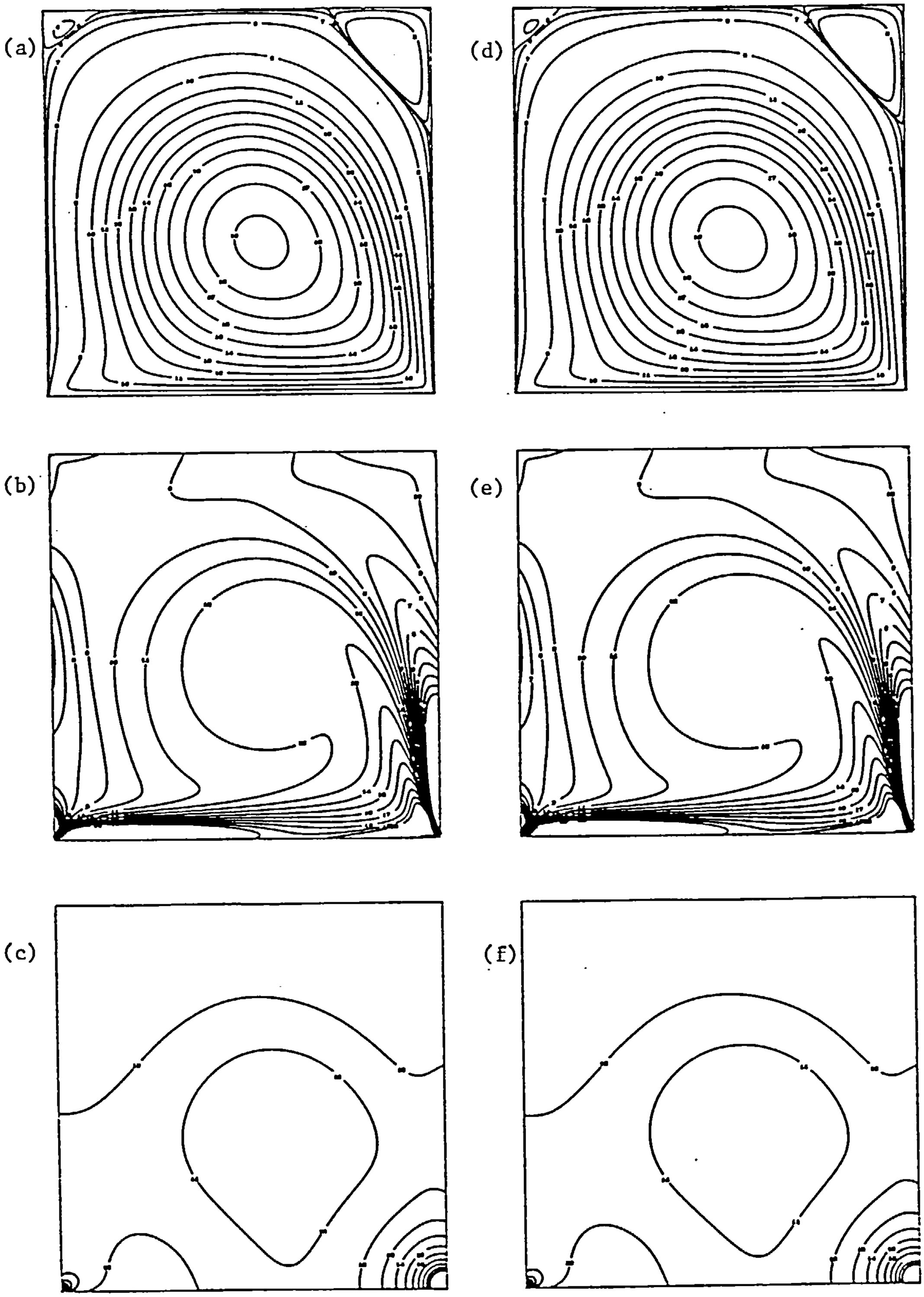


Figure 4.10: Contour plot for Reynolds number 400 for 256^2 internal nodes; hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e), streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

Re	100		400		1 000	
Disc.	hybrid	CCCT	hybrid	CCCT	hybrid	CCCT
ψ_m	0.10317	0.10327	0.11271	0.11417	0.11608	0.12080
x	0.61523	0.61523	0.55273	0.55273	0.52930	0.52930
y	0.26367	0.26367	0.39648	0.39648	0.43555	0.43555
ω_m	3.1490	3.1524	2.2513	2.3098	1.9877	2.1552
ψ_r	-1.186E-5	-1.246E-5	-6.238E-4	-6.728E-4	-1.705E-2	-1.734E-2
x	0.94336	0.94336	0.88867	0.88477	0.86133	0.86133
y	0.93945	0.93945	0.87695	0.87695	0.88867	0.88867
ω_m	-3.381E-2	-3.357E-2	-4.383E-1	-4.782E-1	-1.133	-1.137
ψ_l	-1.916E-6	-1.719E-6	-1.543E-5	-1.303E-5	-2.385E-4	-2.352E-4
x	0.03320	0.03320	0.05273	0.05273	0.08398	0.08398
y	0.96289	0.96680	0.95508	0.95117	0.92383	0.91992
ω_l	-1.587E-2	-1.326E-2	-5.781E-2	-5.320E-2	-3.517E-1	-3.690E-1

Table 4.6: Selected data for the driven cavity with 256^2 node at various Reynolds numbers and with both discretisations.

Chapter 3 (Figure 3.20), CCCT is more efficient at generating solutions of a given accuracy, despite its higher cpu requirement.

Finally, Table 4.7 shows the convergence factor θ, defined by

$$\theta = \left[\frac{\text{final residual}}{\text{initial residual}}\right]^{\frac{1}{PGWU}}. \qquad (4.16)$$

As one would expect, from the results outlined above, the factors obtained with multigridding are much lower than those without. The latter increase as the grid is refined. This is the manifestation of the theory that says that for fine grids, wavelengths that are removed from the mesh size are difficult to eliminate. The multigrid convergence factors decrease (on the whole) as mesh size decreases, and optimality is achieved. Optimal behaviour is achieved at coarser meshes for lower Reynolds numbers, because the information required is not as complex. Linden et al[9] presented results for a hybrid BIM multigrid technique on a non-staggered mesh. This gave convergence factors similar to, but on the whole less than, those presented here. In view of the use here, of staggered grids and the associated problems (see Chapter 7) the factors result from this work compare well. The factors for CCCT, whilst not differing much for Reynolds numbers of 100 and 400, are not as good as those obtained with hybrid at Reynolds number 1 000. This, as explained earlier, is

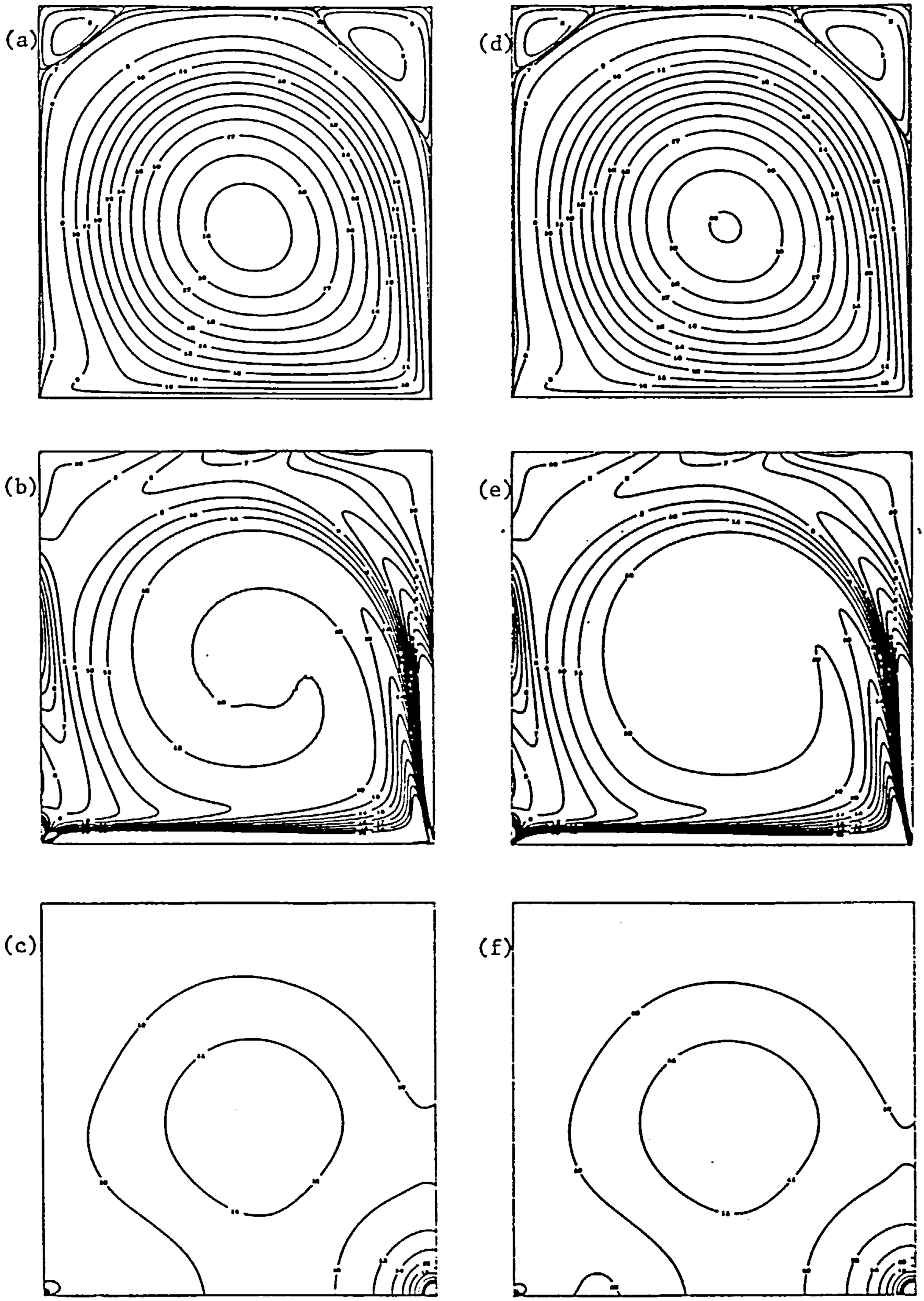


Figure 4.11: Contour plot for Reynolds number 1 000 for 256^2 internal nodes; hybrid (a) - (c), CCCT ($\alpha = 0$) (d) - (e), streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

a) hybrid

Re	100		400		1000	
	NMG	MG	NMG	MG	NMG	MG
4	0.32	0.32	0.40	0.40	0.39	0.39
8	0.45	0.45	0.47	0.60	0.59	0.65
16	0.71	0.33	0.60	0.55	0.79	0.49
32	0.89	0.31	0.79	0.55	0.90	0.66
64	0.96	0.25	0.94	0.53	0.95	0.75
128	0.99	0.18	0.98	0.40	0.99	0.67
256	-	0.22	-	0.31	-	0.42

b) CCCT

Re	100		400		1000	
	NMG	MG	NMG	MG	NMG	MG
4	0.21	0.21	0.38	0.38	0.68	0.68
8	0.43	0.45	0.71	0.74	0.92	0.87
16	0.60	0.3	0.80	0.64	0.96	0.77
32	0.78	0.21	0.88	0.57	0.96	0.74
64	0.93	0.19	0.93	0.43	0.97	0.70
128	0.94	0.19	0.95	0.28	0.97	0.64
256	-	0.20	-	0.30	-	0.65

Table 4.7: Multigrid convergence factor for both discretisations

due to the linearisation involved. CCCT performance could be improved by use of a solution technique that handled these non-linear terms better (see Chapter 7). The sudden drop in the convergence rate for hybrid between meshes of 128^2 and 256^2 was also observed by Linden et al. This is caused by the change to predominant use of central differencing as the cell Reynolds number drops below 2. Upwind differencing, with its inherent mesh-dependent numerical diffusion, limits convergence rates. From Gaskell, Lau and Wright[5] it can be seen that, along the line $x = 0.5$ where velocities are highest, the velocity gives a cell Reynolds number of 2 or less (with a 128^2 grid) from $y = 0.09$ to $y = 1.0$. So central differencing is used throughout at least 90% of the flow region. This means that corrections to regions of the fine-mesh discretised with central differencing are also calculated with central differencing on the coarser mesh. This does not always happen on lower levels and so convergence is degraded. It must be said that this demonstrates that central differencing is well suited to a multigrid approach. However, it is necessary to maintain the cell Reynolds number less than 2 throughout the flow domain, which clearly is totally impractical.

References

1. Brandt, A. and Dinar, N., "Multigrid Solution to Elliptic Flow Problems", pp. 53-147 in *Numerical Methods in Partial Differential Equations*, ed. S. Parter, (1977).

2. Doormaal, J.P. Van and Raithby, G.D., "Enhancements of the SIMPLE Method for Predicting Incompressible Fluid Flows", *Numerical Heat Transfer* 7 pp. 147-163 (1984).

3. Falle, S.A.E.G. and Wilson, M.J., "Multigrid Calculation of Jet Flows", pp. Clarendon Press in *Proceedings of the ICFD Conference on Numerical Methods for Fluids Dynamics - in press*, ().

4. Gaskell, P.H. and Lau, A.K.C., "Curvature Compensated Convective Transport : SMART, a New Boundedness Preserving Transport Algorithm", *International Journal for Numerical Methods in Fluids* 8 pp. 617-641 (1988).

5. Gaskell, P.H., Lau, A.K.C., and Wright, N.G., "Comparison of Two Strategies for Use with Higher Order Discretisation Schemes in Fluid Flow Simulation", *International Journal for Numerical Methods in Fluids - in press*, 0.

6. Gaskell, P.H. and Wright, N.G., "Multigrids Applied to a Solution Technique for Recirculating Fluid Flow Problems", *The Proceedings of the IMA Conference on Simulation and Optimisation of Large Systems*, pp. 51-55 (1988).

7. Hutchinson, B.R., Galpin, P.F., and Raithby, G.D., "Application of Additive Correction Multigrid to the Coupled Fluid Flow Equations", *Numerical Heat Transfer* 13(2)(1988).

8. Leonard, B.P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", *Computational Methods in Applied Mechanics and Engineering* 19 pp. 59-98 (1979).

9. Linden, J., Steckel, B., and Stuben, K., "Parallel Multigrid Solution of the Navier-Stokes Equation on General 2D-Domains", *Arbeitspapiere der GMD*,

(294)(1988).

10. Lonsdale, G., "Solution of a Rotating Navier-Stokes Problem by a Nonlinear Multigrid Algorithm", *International Journal for Numerical Methods in Fluids*, (1988).

11. Miller, T.F. and Schmidt, F.W., "Evaluation of a Multilevel Technique Applied to the Poisson and Navier-Stokes Equations", *Numerical Heat Transfer* **13** pp. 1-26 (1988).

12. Patankar, S.V. and Spalding, D.B., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three-Dimensional Parabolic Flows", *International Journal of Heat Mass Transfer* **15** pp. 1787-1806 (1972).

13. Phillips, R.E. and Schmidt, F.W., "Multigrid Techniques for the Solution of the Passive Scalar Advection-Diffusion Equation", *Numerical Heat Transfer* **8** pp. 25-43 (1985).

14. Phillips, R.E. and Schmidt, F.W., "A Multilevel-Multigrid Technique for Recirculating Flows", *Numerical Heat Transfer* **8** pp. 573-594 (1985).

15. Raithby, G.D., "Skew Upstream Differencing Schemes for Problems Involving Fluid Flow", *Computer Methods in Applied Mechanics and Engineering* **9** pp. 153-164 (1976).

16. Settari, A. and Aziz, K., "Generalization of the Additive Correction Methods for the Iterative Solution of Matrix Equations", *SIAM Journal of Numerical Analysis* **10** pp. 56-521 (1971).

17. Shaw, G. and Sivaloganathan, S., "On the Smoothing Properties of the SIMPLE Pressure-Correction Algorithm", *International Journal for Numerical Methods in Fluids* **8** pp. 441-461 (1988).

18. Sivaloganathan, S. and Shaw, G., "A Multigrid Method for Recirculating Flows", *International Journal for Numerical Methods in Fluids*, (1988).

19. Spalding, D.B., "A Novel Finite Difference Formulation for Differential Expressions Involving both First and Second Derivatives", *International Journal for Numerical Methods in Engineering* **4** pp. 551-559 (1972).

Chapter 5

EXTENSION OF METHOD TO MORE COMPLEX SITUATIONS

5.1. Introduction

In the Chapter 4 an efficient solution strategy for simulating recirculating flows was outlined. It consists of a multigrid algorithm and unsegregated solver and was shown to be capable of producing solutions to the problem of flow in a lid-driven cavity, on very fine meshes. The problem examined is of interest because of the large areas of recirculation contained by the flow. It would be premature, at this stage, to suggest that the problem was representative of all flows of this type, which may possess more complicated features. In order to assert the usefulness of the method more fully we must apply it to more varied flow situations.

One important aspect of many real life problems is that they are best posed in a three-dimensional co-ordinate system. Two-dimensional or axisymmetric simplifications cannot always fully capture the features inherent in them. However, the main drawback to extending solution techniques to three dimensions is the order of magnitude increase in the number of finite difference nodes required, which leads to a corresponding increase in cpu time. With traditional solvers, where computer time increases exponentially with the number of nodes, this can be quite prohibitive. However, this problem is not encountered with multigrid methods, and so three dimensional solvers are feasible. In view of this it was decided to extend the method to three-dimensions.

The first problem considered here is one of a three-dimensional lid-driven cavity - an obvious extension of the problem considered earlier. Solution of this flow field is facilitated by the existence of an axis of symmetry (see Figure 5.1(a)). Consequently, only half the cavity need be solved - a derivative boundary condition being applied at the plane of symmetry. This leads us to examine a further extension of the multigrid algorithm, namely, the treatment of derivative boundary conditions.

The second test case considered involves a more rigorous test of our treatment of derivative conditions. The sudden expansion of a plane channel has a derivative condition at the the outflow. This has caused problems in the past for researchers

attempting to use multigrids for this type of flow - it is even problematic for non-multigrid solution procedures. For multigrids to be accepted by Computational Fluid Dynamicists in general, such problems must be overcome, because outflows conditions are often encountered in flow situations. Multigrids offer the advantage of fine mesh resolution, that such flows require.

These two problems form a significant extension of the algorithm presented in Chapter 4.

5.2. The Three Dimensional Lid Driven Cavity

The equations for three dimensional incompressible flow in cartesian coordinates are

$$u_j \frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j^2}, \quad i=1,2,3$$

$$\frac{\partial u_j}{\partial x_j} = 0$$

The grid arrangement is staggered in a manner analogous to that in two-dimensions, with a pressure or scalar node positioned at the centre of each cubic control volume and a velocity perpendicular to and at the centre of each face (see Figure 5.1(b)). Each of the above equations is integrated over this control volume to form an algebraic set of equations.

The flow configuration for this test case is given in Figure 5.1(a). The moving wall is in the plane y=0, where the velocity component w is equal to one. The remaining velocities on this wall and all the velocities on each of other the walls are set equal to zero, except that is, at the plane of symmetry. Here, u is is set to zero together with the derivatives $\frac{\partial v}{\partial x}$ and $\frac{\partial w}{\partial x}$.

Some researchers have found difficulty extending their solution technique to three dimensions, because of algorithmic complications. This problem was not encountered with the smoothing technique employed here. The Block Implicit Method is easily

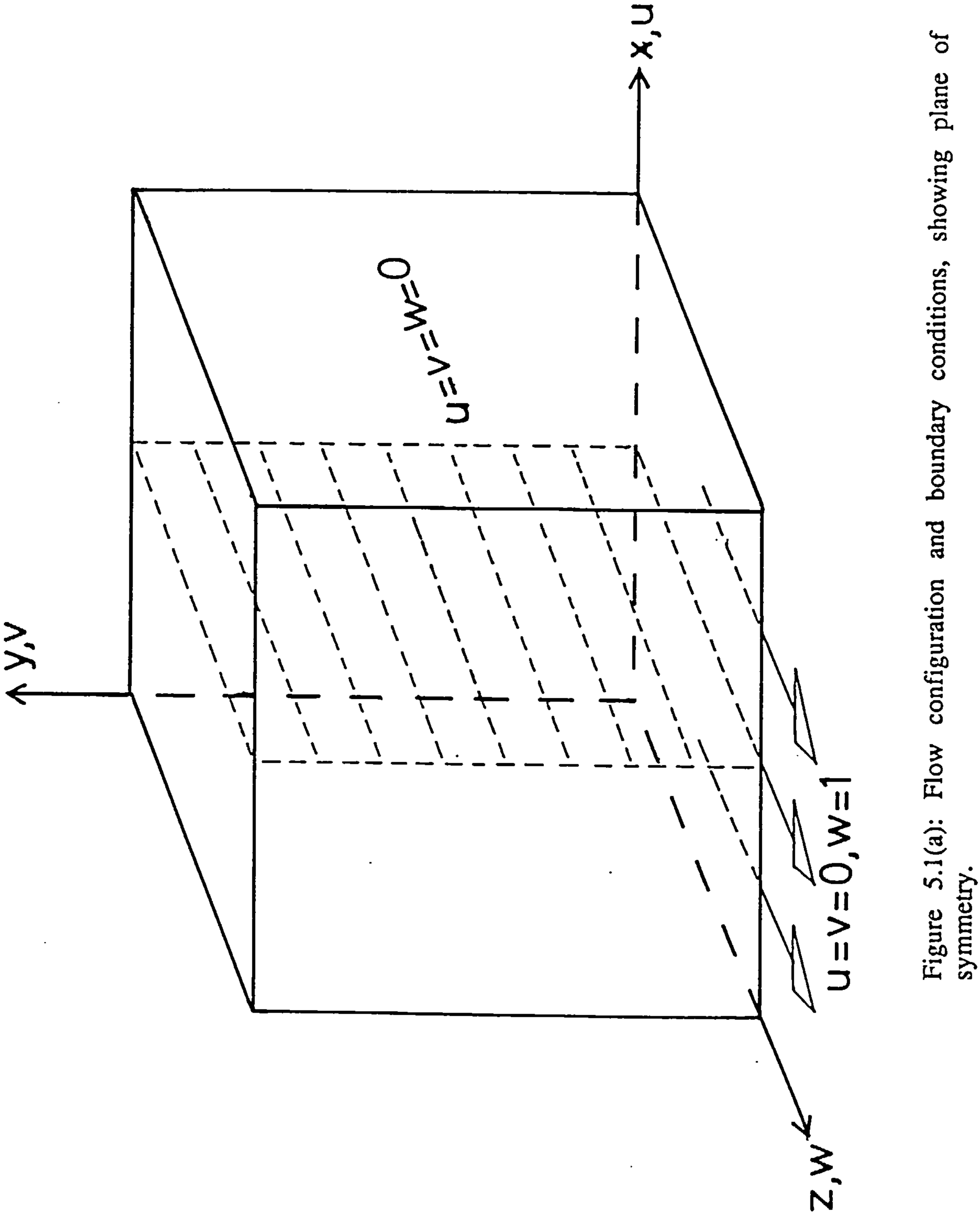


Figure 5.1(a): Flow configuration and boundary conditions, showing plane of symmetry.

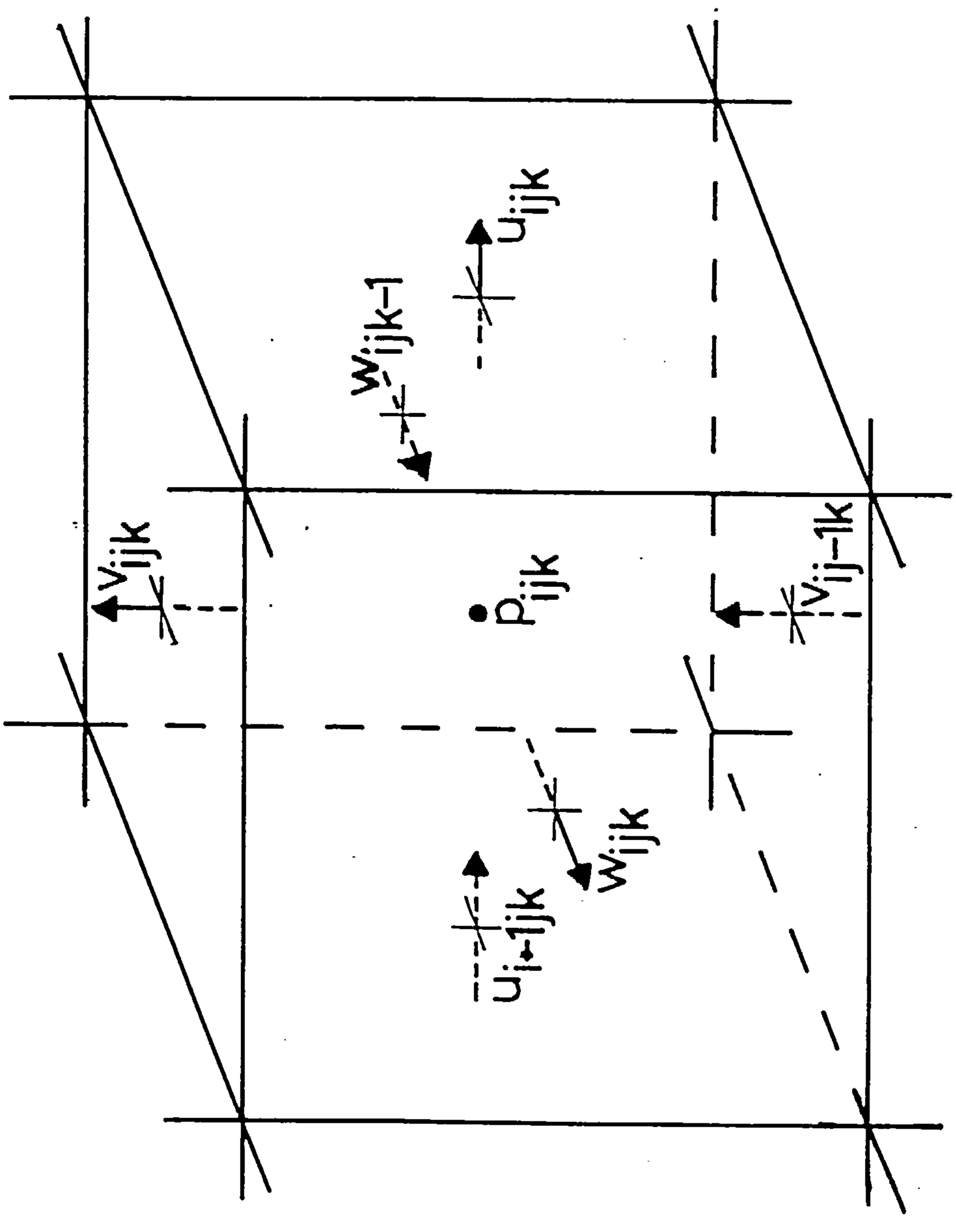


Figure 5.1(b): Three dimensional control volume.

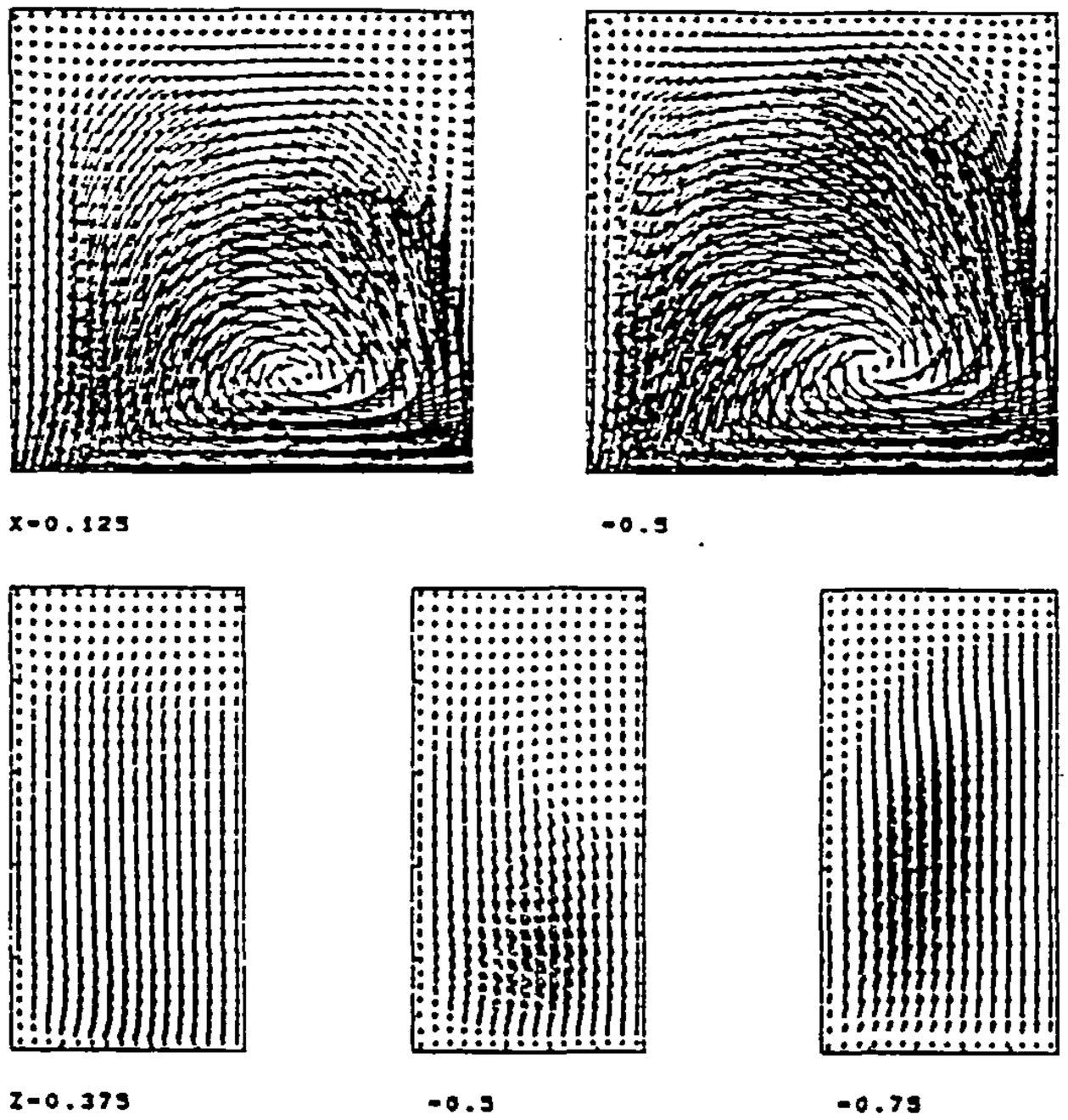


Figure 5.2: Velocity vectors obtained with hybrid for Reynolds number 100.

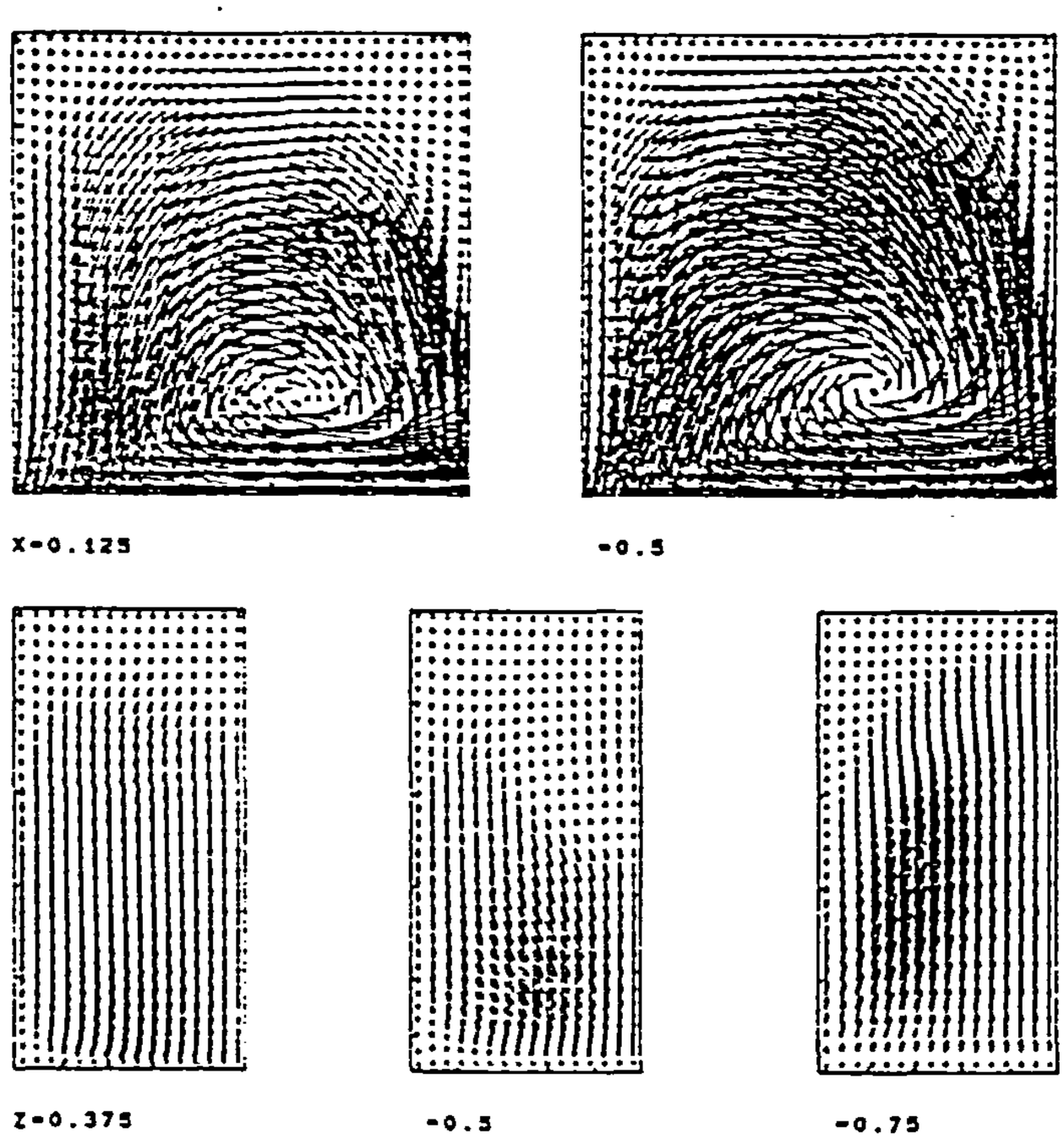


Figure 5.3: Velocity vectors obtained with CCCT ($\alpha = 0$) for Reynolds number 100.

modified for a three dimensional domain. Each control volume is visited in turn, the six velocities and one pressure associated with it being updated simultaneously. The system generated by the equations for these variables form a 7x7 coefficient matrix of a similar nature to that generated in two dimensions. Once again, this is easily solved using LU-decomposition (see Appendix II). As before the corrections are multiplied by relaxation factors α_u and α_p, after they have been calculated.

As we saw in Chapter 3 an important feature of the Block Implicit Method is the significance of the order in which the cells are visited. The procedure adopted therefore allows for four sweeps within one iterative cycle in a manner similar to before. That is

(1) i increasing j increasing k increasing

(2) i decreasing j decreasing k decreasing

(3) j increasing i increasing k increasing

(4) j decreasing i decreasing k decreasing

As in Chapter 3 it was considered undesirable to carry out too many sweeps within one iteration as this would make the smoother inefficient on fine grids.

The derivative boundary condition is implemented in an explicit manner. During one iteration the present values are assumed fixed and used in

the calculation of updates. After the iteration has been carried out the boundary values are updated to satisfy the boundary conditions. If hybrid[10] differencing is used a linear profile is assumed over the boundary layer, that is, if ϕ_N is the boundary value and ϕ_{N-1} the value adjacent to the boundary (see Figure 3.6(a)), we assume that

$$\phi(x)=a+bx.$$

So

$$\phi_N=a$$

and

$$\phi_{N-1}=a-\frac{bh}{2}.$$

Therefore

$$a=\phi_N$$

and

$$b=\frac{2}{h}(\phi_N-\phi_{N-1}).$$

Hence

$$\phi'(0)=0,$$

which implies that

$$\phi_N=\phi_{N-1}.$$

If CCCT(α=0)[5] is used a quadratic profile is assumed, namely

$$\phi(x) = a+bx+cx^2,$$

giving

$$\phi_N = a,$$

and

$$\phi_{N-1} = a-\frac{bh}{2}+\frac{ch^2}{4},$$

$$\phi_{N-2} = a-\frac{3bh}{2}+\frac{9ch^2}{4}.$$

So

$$a = \phi_N,$$

and

$$b = \frac{1}{3h}(\phi_{N-2}-9\phi_{N-1}+8\phi_N).$$

Hence

$$\phi'(0)=0,$$

implying that

$$\phi_N=\frac{1}{8}(9\phi_{N-1}-\phi_{N-2}).$$

These formulae correspond to the use of an 'image' point as proposed by Leonard[6]. All other boundary conditions are treated as before.

When multigridding, special attention must be paid to the treatment of derivative conditions. Incorrect implementation can lead to degradation of convergence and even completely annul the advantages of multigrids. The procedure used here is as follows.

(i) When the fine grid values are restricted onto the coarse grid, the boundary values are also restricted using linear interpolation.

(ii) These boundary values are then updated during the smoothing process on the coarse grid.

(iii) When the correction is prolonged, a correction to the boundary values is also prolonged using linear interpolation.

(iv) The fine grid is then smoothed with updating of boundaries.

The reasoning behind this strategy is covered in a discussion of derivative boundary conditions later in this chapter. Great care must be taken when programming such algorithms since even a single "bug" can completely destroy multigrid convergence, while still producing the correct solution.

The restriction and prolongation of information is carried via linear interpolation in a manner analogous to the two-dimensional case. The only exception is that at a boundary where a zero derivative condition is specified, near boundary values are updated using the boundary values themselves. In other respects the implementation of the multigrid is exactly the same as for the two-dimensional problem.

5.3. Results

Results were obtained for Reynolds numbers of 100 and 1 000 with a grid system containing upto 32 x 32 x 16 internal nodes for both hybrid and CCCT differencing.

The residual definition is analogous with that in two-dimensions and the tolerance is again taken as $5x10^{-5}$. This was found to be the largest number of nodes that it was possible to employ within the memory restrictions of our computer.

Salient results are presented in Tables 5.1 to 5.5 Figures 5.2 to 5.12. From Figures 5.2 to 5.5, showing velocity vectors, it is interesting to compare the two solutions obtained with hybrid and CCCT(α=0) differencing. CCCT resolves the eddy in the top left of 5.5(d) at Reynolds number of 1 000 much better than hybrid. This has also been observed by Gaskell and Lau[4] for various flows containing one or more recirculating zones. Figure 5.6 show a comparison of the centreline velocities for the flow at Reynolds numbers of 100 and 1 000. Although these are practically indistinguishable in the former case, they show significant differences in the latter. Figures 5.7 to 5.8 for the streamfunction and vorticity on the central plane also highlight the differences between the two solutions. At Reynolds number 1 000 the predicted flows are significantly different, due to the numerical diffusion inherent in hybrid. Such differences, on what may be considered to be a relatively fine mesh for a three-dimensional problem, demonstrates the benefits of using a higher order convective transport approximation. An advantage that is gained only at the expense of a slight increase in computer time. The figures show results consistent with those obtained for the two-dimensional cavity on the same meshes. The data in Table 5.1, for streamfunction maxima and minima, shows that these values are less then those predicted in the two-dimensional simulations - see Tables 3.5 and 3.6. This is due to the retarding effect of the side walls.

The relaxation factors are given in Table 5.2. From Table 5.3 it can be seen that the computer time is proportional to the number of nodes and that h-independence has been achieved. These times are lower compared with those for BIM and SIMPLE (Table 5.4) and show that multigrids make the solution of three-dimensional flow easily realisable. Figures 5.9 to 5.12 show plots of log R against FGWU. These are predominantly straight lines, apart that is from some minor deviations as the residual approaches the tolerance for Reynolds number 1 000. The number of levels used is

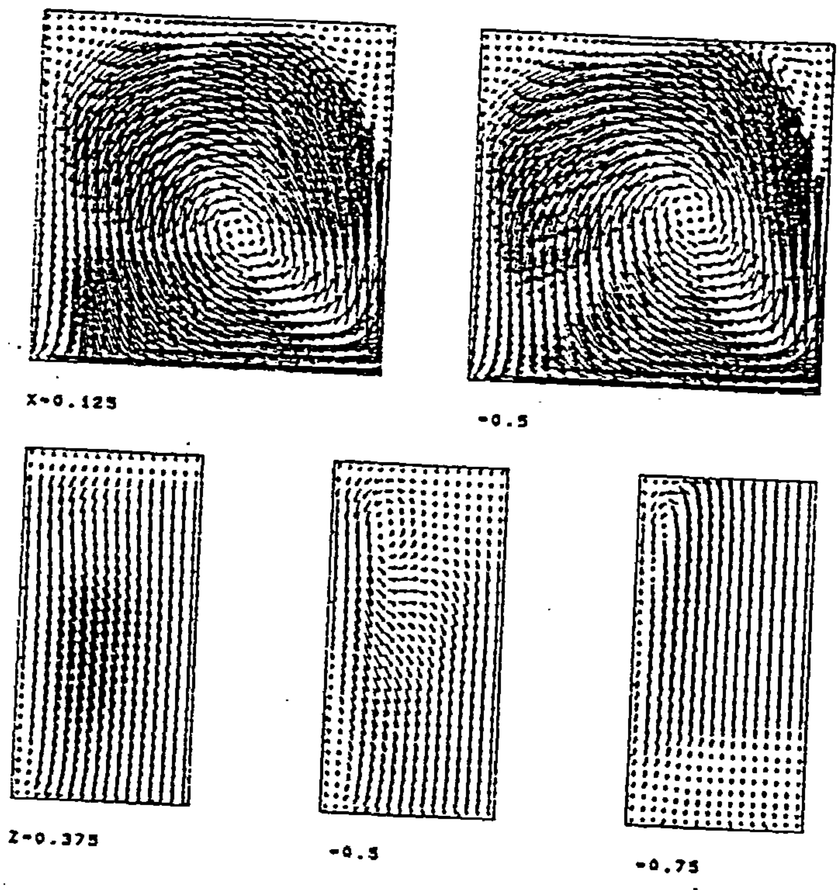


Figure 5.4: Velocity vectors obtained with hybrid for Reynolds number 1 000.

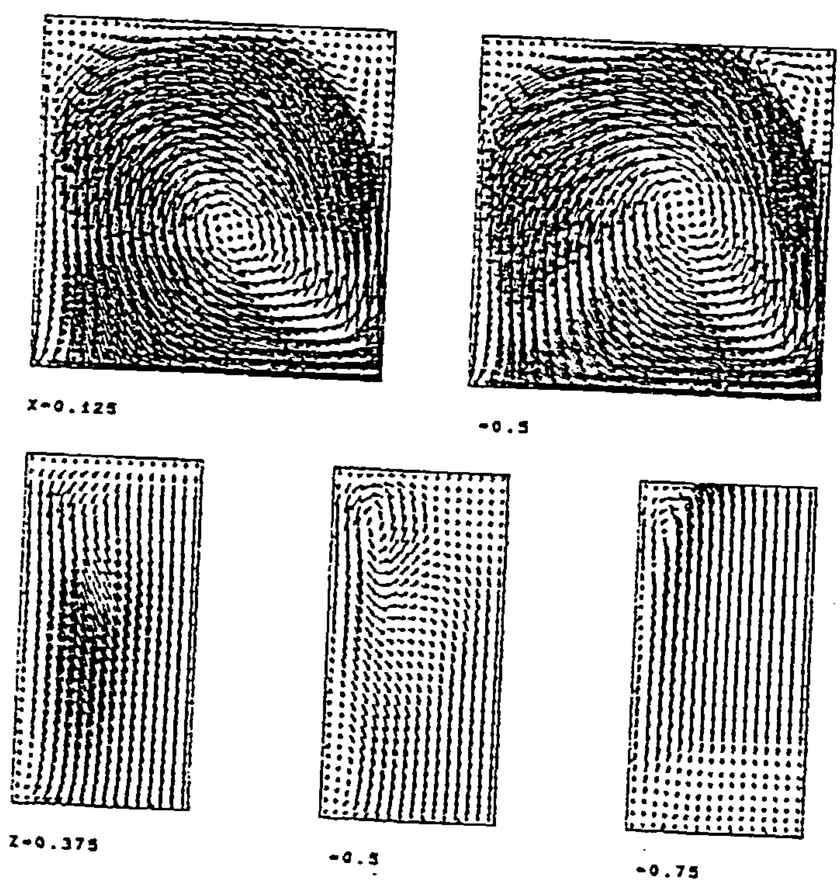


Figure 5.5: Velocity vectors obtained with CCCT ($\alpha = 0$) for Reynolds number 1 000.

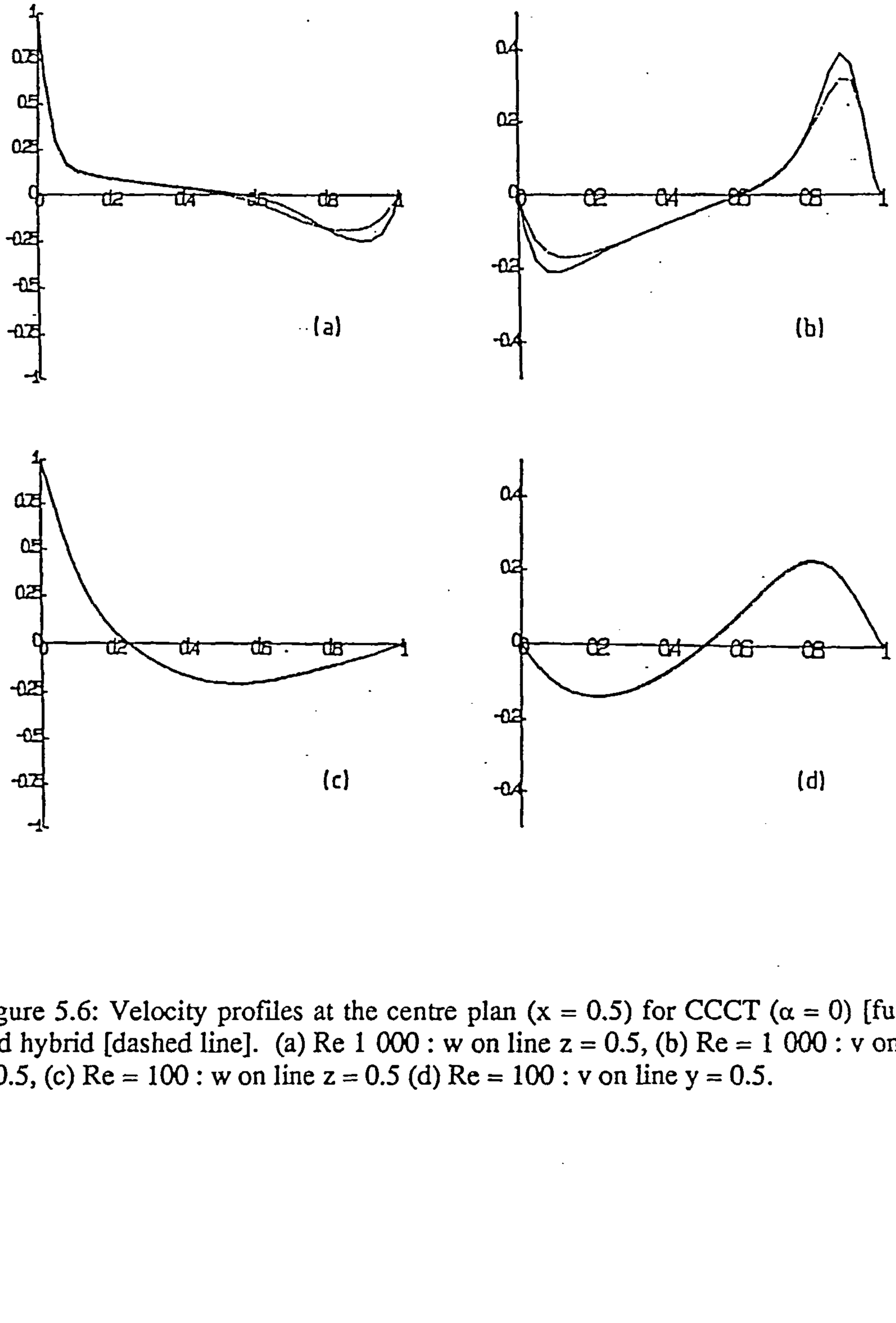


Figure 5.6: Velocity profiles at the centre plan (x = 0.5) for CCCT (α = 0) [full line] and hybrid [dashed line]. (a) Re 1 000 : w on line z = 0.5, (b) Re = 1 000 : v on line y = 0.5, (c) Re = 100 : w on line z = 0.5 (d) Re = 100 : v on line y = 0.5.

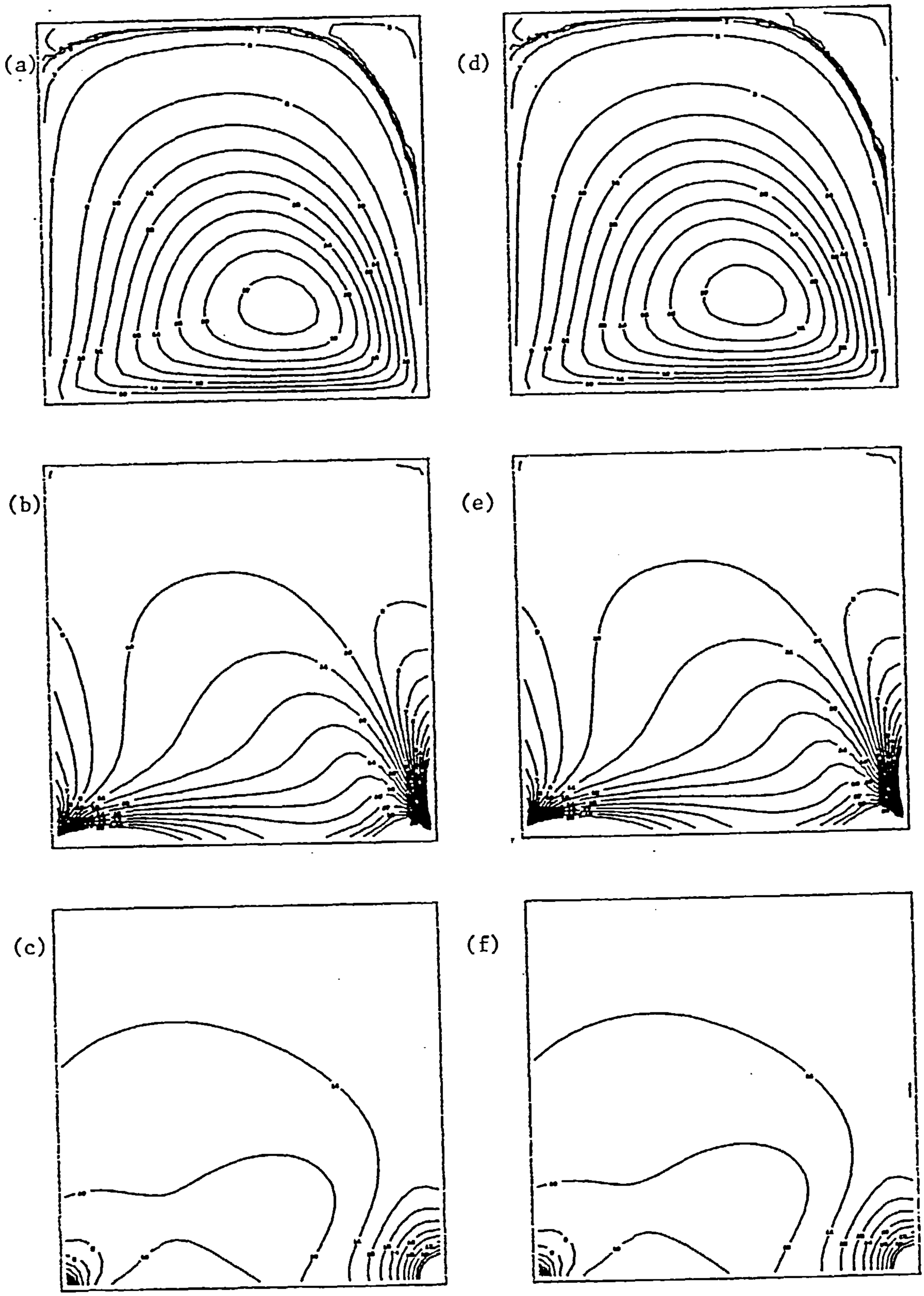


Figure 5.7: Contour plots on central plane (x = 0.5) for Reynolds number 100 using 32x32x16 internal nodes: hybrid (a) - (c), CCCT (α = 0) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

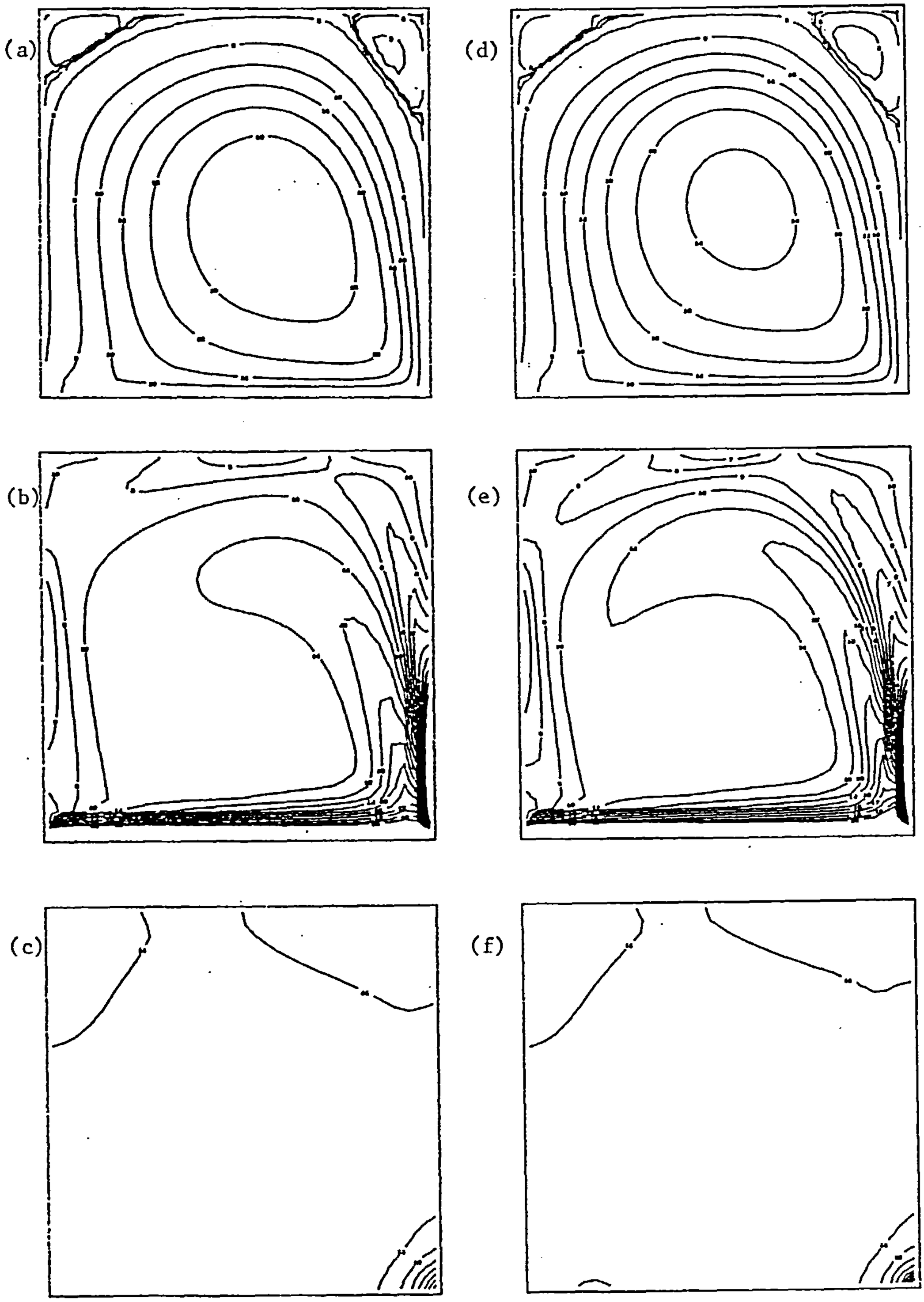


Figure 5.8: Contour plots on central plane (x = 0.5) for Reynolds number 1 000 using 32x32x16 internal nodes: hybrid (a) - (c), CCCT (α = 0) (d) - (e); streamfunction (a) and (d), vorticity (b) and (e) and pressure (c) and (f).

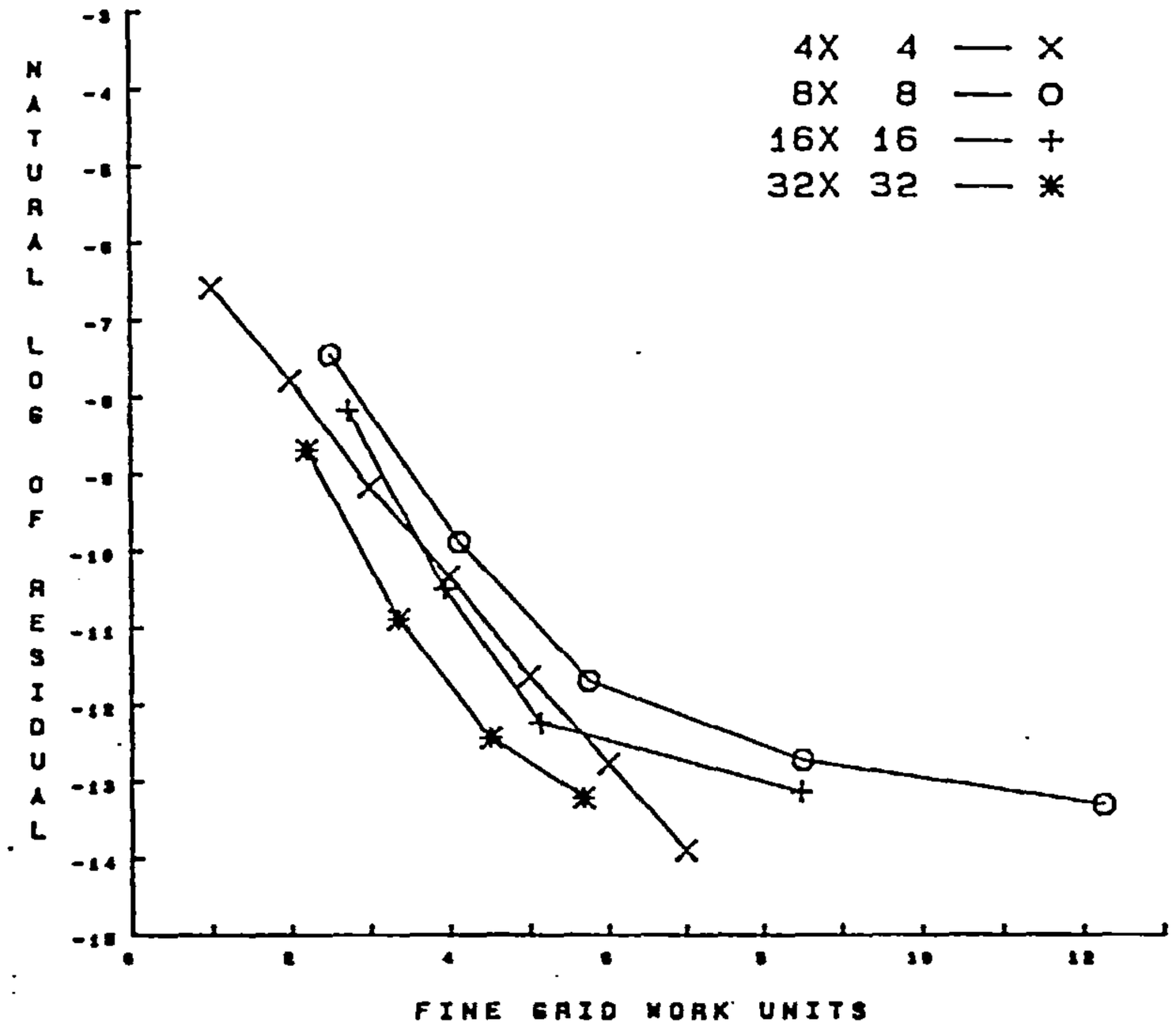


Figure 5.9: Plot of the natural log of the residual against FGWU for Reynolds number 100 and hybrid differencing.

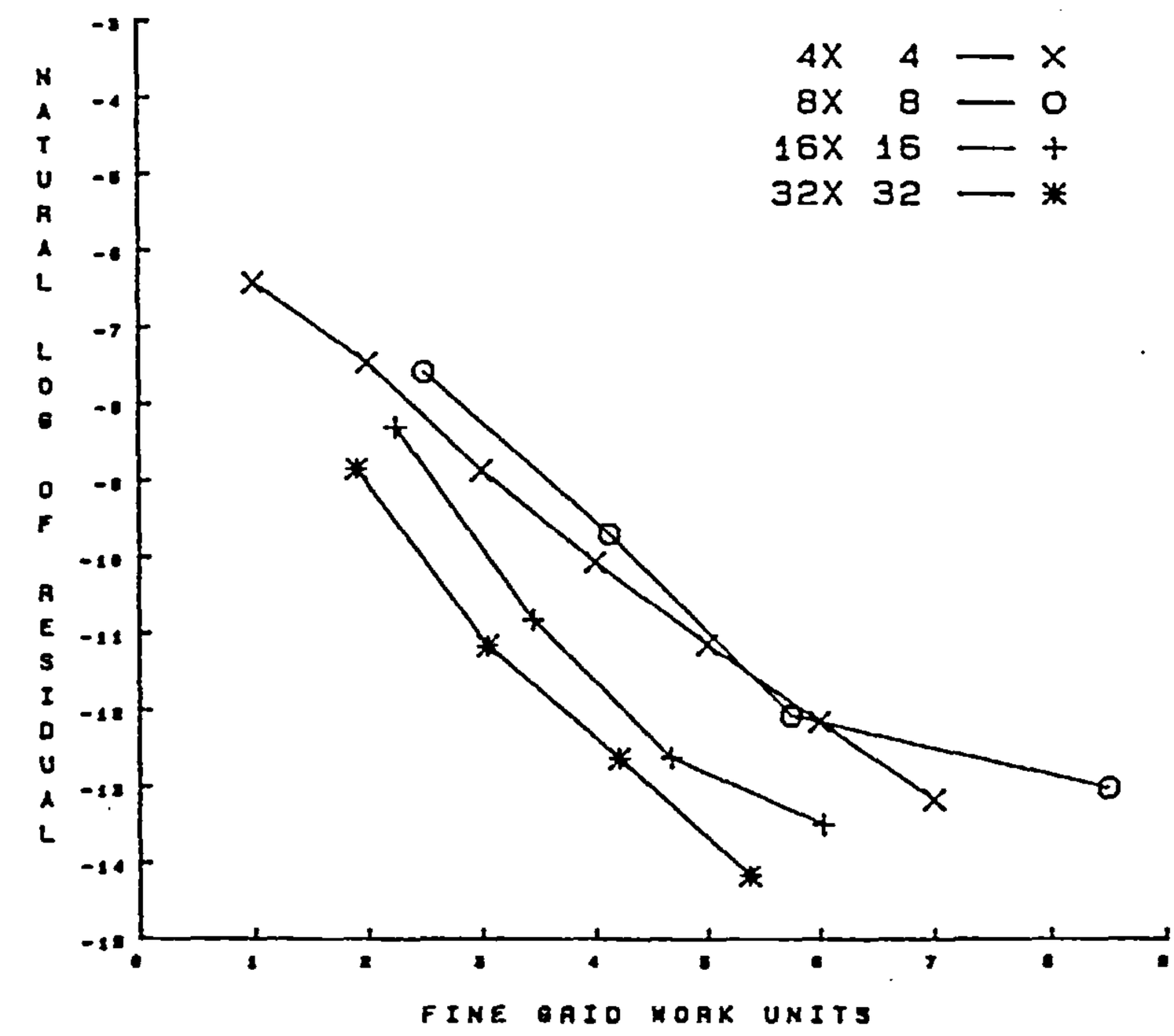


Figure 5.10: Plot of the natural log of the residual against FGWU for Reynolds number 100 and CCCT ($\alpha = 0$) differencing.

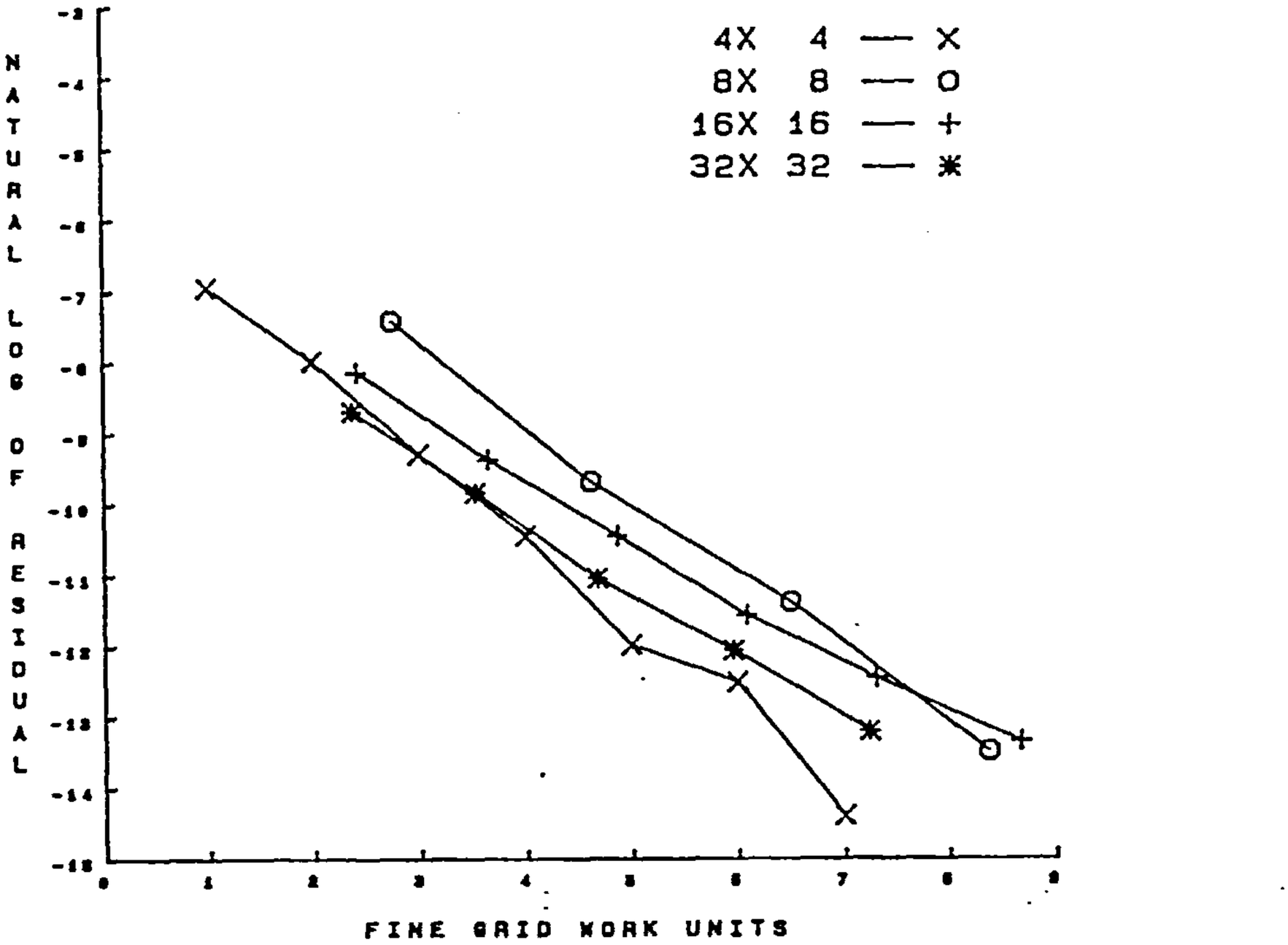


Figure 5.11: Plot of the natural log of the residual against FGWU for Reynolds number 1 000 and hybrid differencing.

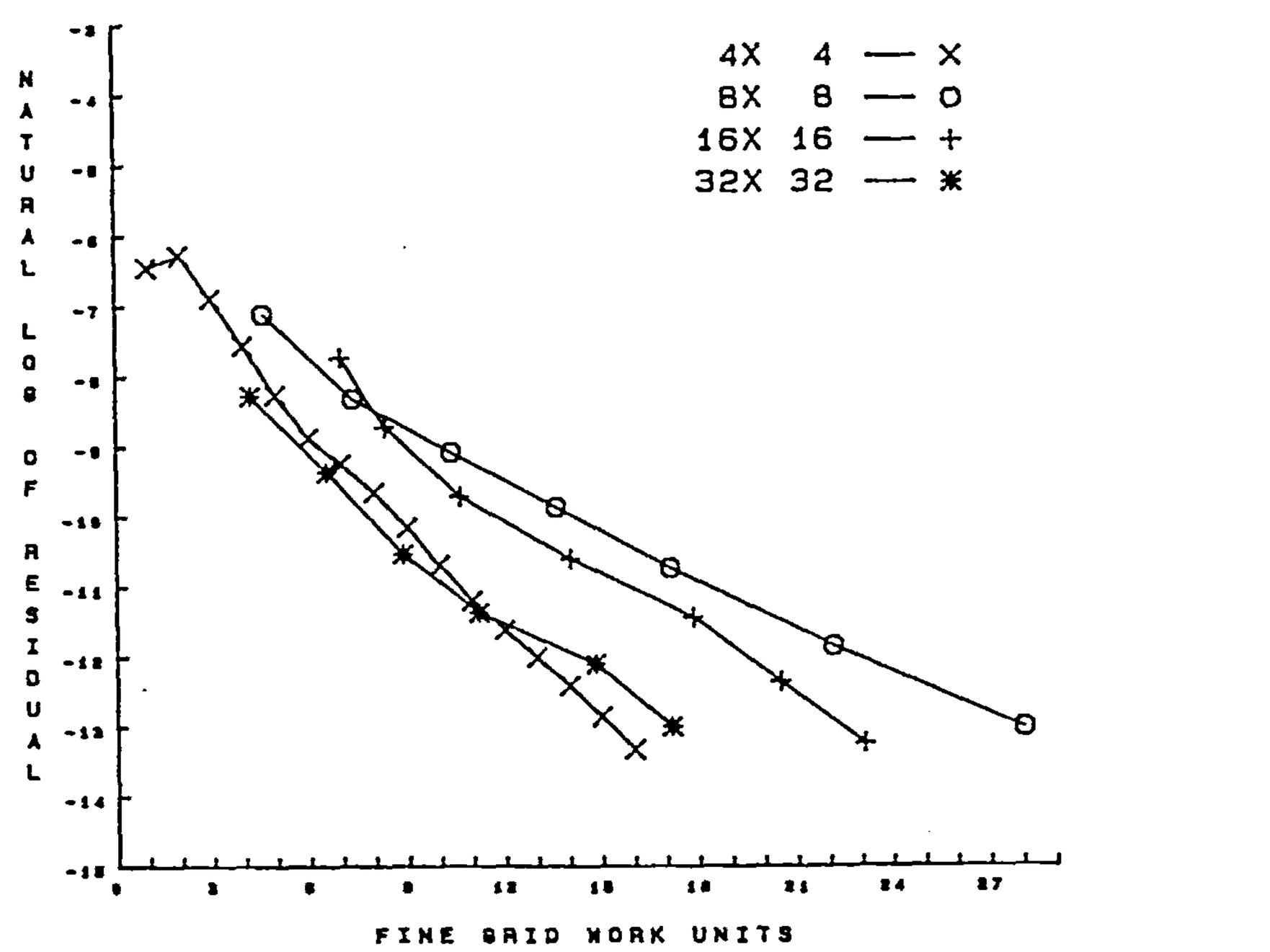


Figure 5.12: Plot of the natural log of the residual against FGWU for Reynolds number 1 000 and CCCT ($\alpha = 0$) differencing.

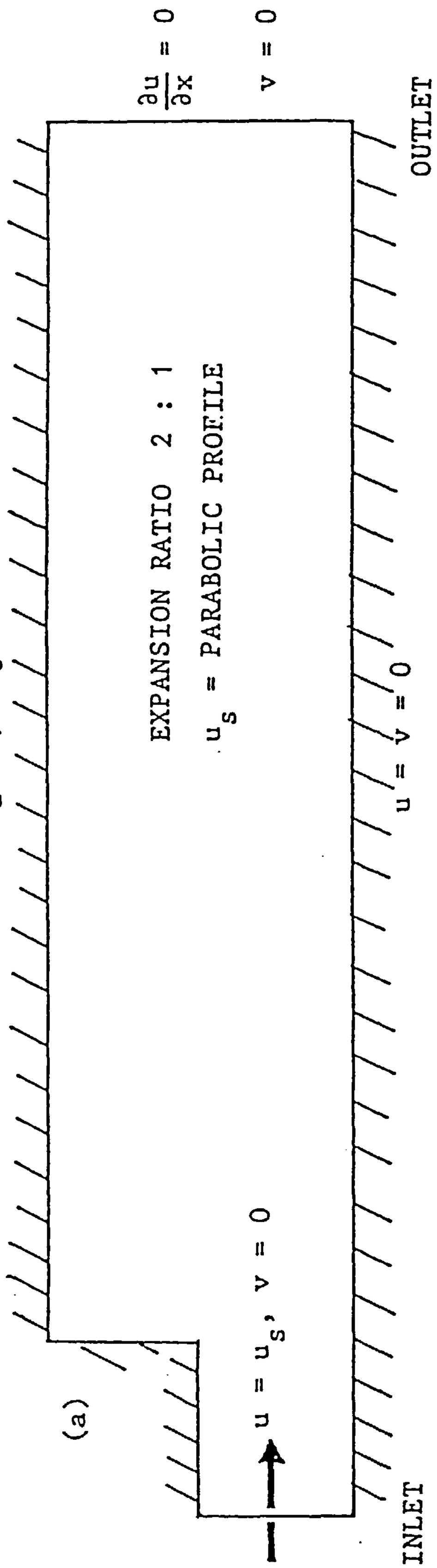


Figure 5.13: Flow configuration for the sudden expansion problem.

Re	100		1 000	
Disc.	hybrid	CCCT	hybrid	CCCT
ψ_m	0.09520	0.09552	0.05933	0.06345
x	0.60938	0.60938	0.60938	0.57813
y	0.23438	0.23438	0.45313	0.4438
ω_m	2.8547	2.8457	0.65841	0.61654
ψ_l	-4.3376E-5	-4.9384E-5	-3.7386E-4	-5.2424E-4
x	0.20313	0.20313	0.10938	0.10938
y	0.98438	0.98438	0.92188	0.92188
ω_l	-0.26656	-0.27657	-0.43824	-0.61021
ψ_r	-3.8767E-4	-4.3010E-4	-1.4111E-3	-1.8898E-3
x	0.92188	0.92188	0.89063	0.89063
y	0.85938	0.85938	0.89063	0.89063
ω_r	-0.19746	-0.20151	-1.2666	-1.0940

Table 5.1: Selected data for the three-dimensional lid-driven cavity for both discretisations and Reynolds numbers.

	100	1 000
hybrid	(0.8,1.3)	(0.6,1.4)
CCCT	(1.0,1.3)	(0.2,1.6)

Table 5.2: Relaxation factors for hybrid and CCCT(α=0) at Reynolds numbers of 100 and 1 000.

Re	100				1 000			
Disc.	hybrid		CCCT		hybrid		CCCT	
Cpu	FGWU	Cpu	FGWU	Cpu	FGWU	Cpu	FGWU	
4x4x2	0.22	7.00	0.25	7.00	0.22	7.60	0.56	16.00
8x8x4	3.54	12.25	2.91	8.50	2.44	8.38	8.84	28.00
16x16x8	20.52	8.48	17.81	6.03	21.55	8.67	61.19	23.09
32x32x16	114.74	5.68	129.18	5.38	144.81	7.24	383.53	17.16

Table 5.3: Computer time(secs.) and FGWU for the three-dimensional lid-driven cavity.

Scheme	BIM	SIMPLE	MG
16x16x8	100.25	75.20	61.19
32x32x16	2578.24	1430.01	383.53

Table 5.4: Comparison of computer times for the Block Implicit Method, SIMPLE and the Multigrid technique presented here, with CCCT(α=0) and at Reynolds number of 1 000.

much smaller than is the case for the two-dimensional cavity, so that multigridding is not optimal which in itself may contribute to the degradation of convergence. Table 5.5 shows the multigrid convergence rate (defined in equation 4.16) for this test case.

Re	100		1 000	
Disc.	hybrid	CCCT	hybrid	CCCT
4x4x2	0.42	0.26	0.43	0.70
8x8x4	0.51	0.33	0.39	0.75
16x16x8	0.37	0.24	0.42	0.75
32x32x16	0.25	0.33	0.39	0.65

Table 5.5: Multigrid convergence rates for the three-dimensional lid-driven cavity for the cases examined here.

They improve as the number of levesl gets larger, indicating that if the use of more levels was possible, then the rates would be even lower. Comparing these rates with those in Chapter 4, it can be seen that they are slightly lower. This would suggest that the incorporation of three-dimensions and a derivative boundary condition has had no adverse effect on the convergence of the multigrid technique used here.

5.4. Flow Through a Sudden Expansion

The geometry of this test problem is shown in Figure 5.13. The length of the channel is 16 times the width of entry. This allows sufficient distance for a physical flow to have achieved uniform conditions. The entry velocity is specified as a parabolic profile with

$$\int_0^{1/16} u \, dy = 1$$

The step itself is located at a distance of 1/16 from the entry and the expansion ratio is 2.

The most salient aspect with regard to multigridding, is the outflow condition. When Fuchs[3] looked at this flow configuration he got around the problem by specifying a parabolic velocity at the outflow. This, however, does not give the same results as for a derivative condition and is unsatisfactory. Vanka[11] solved the problem of the three-dimensional lid-driven cavity described above without making use of symmetry, thus not requiring the use of a derivative condition. Clearly, in a general sense this is an unsatisfactory approach and in a subsequent paper[12] he rectified this

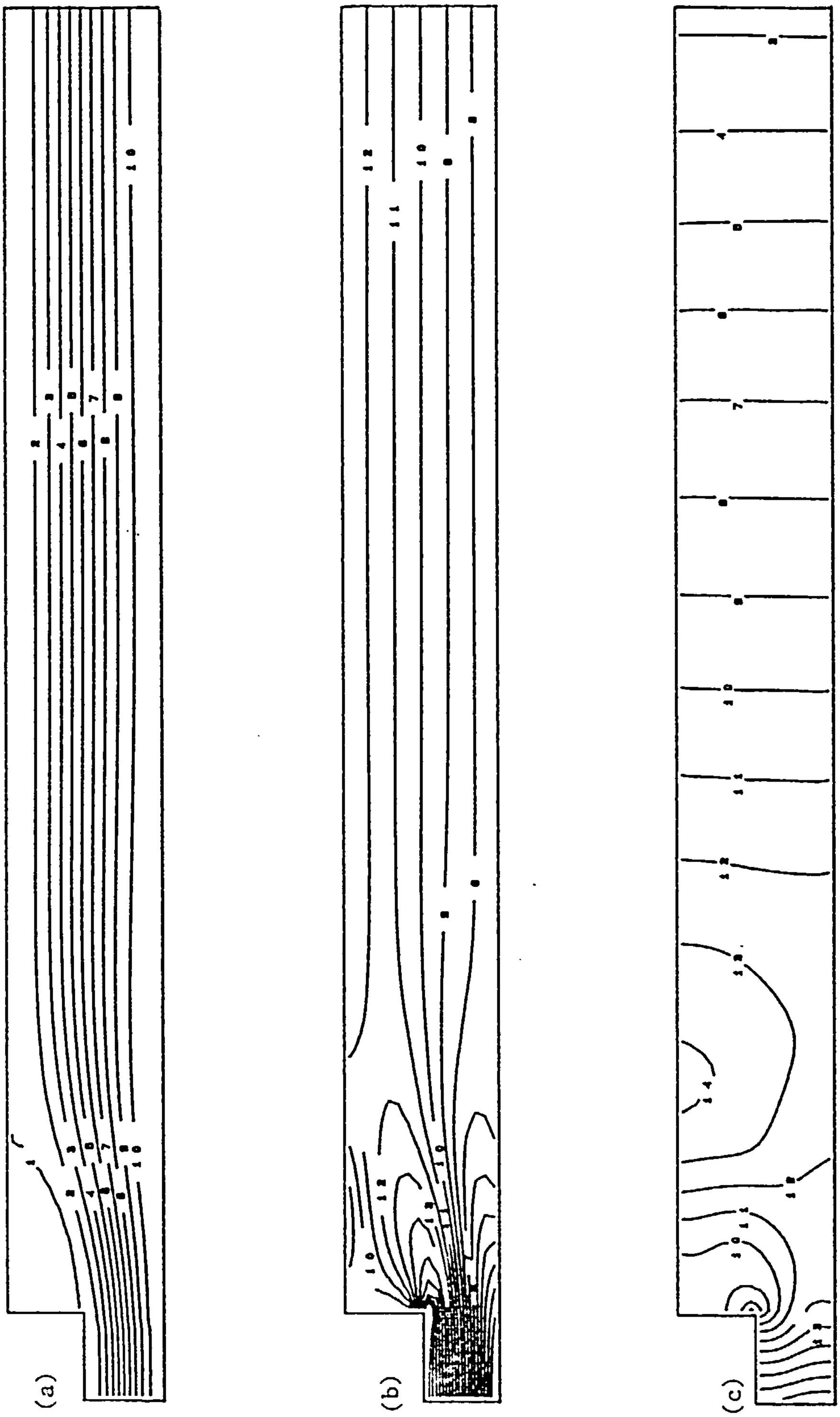


Figure 5.14: Contour plots for flow in a sudden expansion at Reynolds number 50 : (a) streamfunction, (b) vorticity and (c) pressure.

shortcoming by implementing an outflow derivative boundary condition, with a three-dimensional version of the sudden expansion problem. He did this by restricting the boundary conditions onto a coarser mesh and using these as fixed conditions **not** derivative ones. He then prolonged only the interior values onto the fine grid. In order to do this successfully he had to impose the global mass conservation equation

$$\int u.ds = 0$$

on the fine grid. This amounts to scaling the outflow to equal the inflow. Failure to do this would create a set of coarse grid equations with an inherent mass error which would be insoluble with fixed conditions. Vanka's results were quick, but did not show h-independence. The exponential factor, β (cpu αN^{β}) was found to be of the order of 1.3.

The use of fixed boundary values on coarse grids is an erroneous concept carried over from linear multigrids where coarse grid equations are to solve directly for a correction. In non-linear FAS algorithms the coarse grid equations solve for the full variables and thus derivative conditions should be applied. This has been confirmed by other researchers[9,2], and borne out by the work of the author, where more optimal multigridding was achieved with the correct formulation.

In the present work scaling is used to impose global mass conservation, and derivative conditions are imposed on all grids as seen in the previous test problem. The near-boundary nodes were fixed a distance $\frac{h}{2}$ from the boundary and a form of image point was used to impose the conditions (see sub-section 3.5.1). With the outflow velocity the near value is at a distance h and a value occurs on the boundary, so no image point can be used. The condition is therefore imposed directly over a distance h. Obviously, on different grids this length varies and is equivalent to representing a different problem on each grid. Unfortunately, this inconsistency is unavoidable with a staggered grid configuration.

Apart from those mentioned above, the details of the implementation for the sudden expansion are more or less identical to previous cases. The residual is as defined in equation (3.33) and the tolerance is 10^{-5}. The relaxation factors (α_u,α_p) are (0.1,1.6). In view of comparative performance of the two difference schemes it was decided to make use of CCCT only.

Solutions were only obtained for a Reynolds number of 50 - defined as

$$\frac{U_{av}L}{\nu}$$

where U_{av} is the average velocity at the inlet, L is the width of the inlet and ν is the kinematic viscosity. The resulting computer times are presented in Table 5.6.

Solutions at higher Reynolds numbers proved difficult to obtain. This is due to the point-by-point nature of the solver, which necessitates very low relaxation factors.

The results (see Figure 5.14) obtained compare well with those of a comparison exercise[1], with results within a few percent of those given for reattachment length. Multigrid convergence is not achieved in full (see Figure 5.15), but solution times are much less than could be achieved with an ordinary technique. This degradation is a consequence of several factors.

(i) The outflow condition is not properly dealt with. In particular the treatment of the velocity perpendicular to the outflow plane is unsatisfactory, due to the inconsistency between grids.

(ii) There is a singularity at the corner of the expansion. The continuum

Mesh	Cpu.(secs.)	FGWU
16x1	1.62	80.0
32x2	6.96	121.0
64x4	23.76	99.24
128x8	87.04	97.08
256x16	398.72	110.20

Table 5.6: Computer time and FGWU for the sudden expansion at Reynolds of 50 and with CCCT(α=0) differencing.

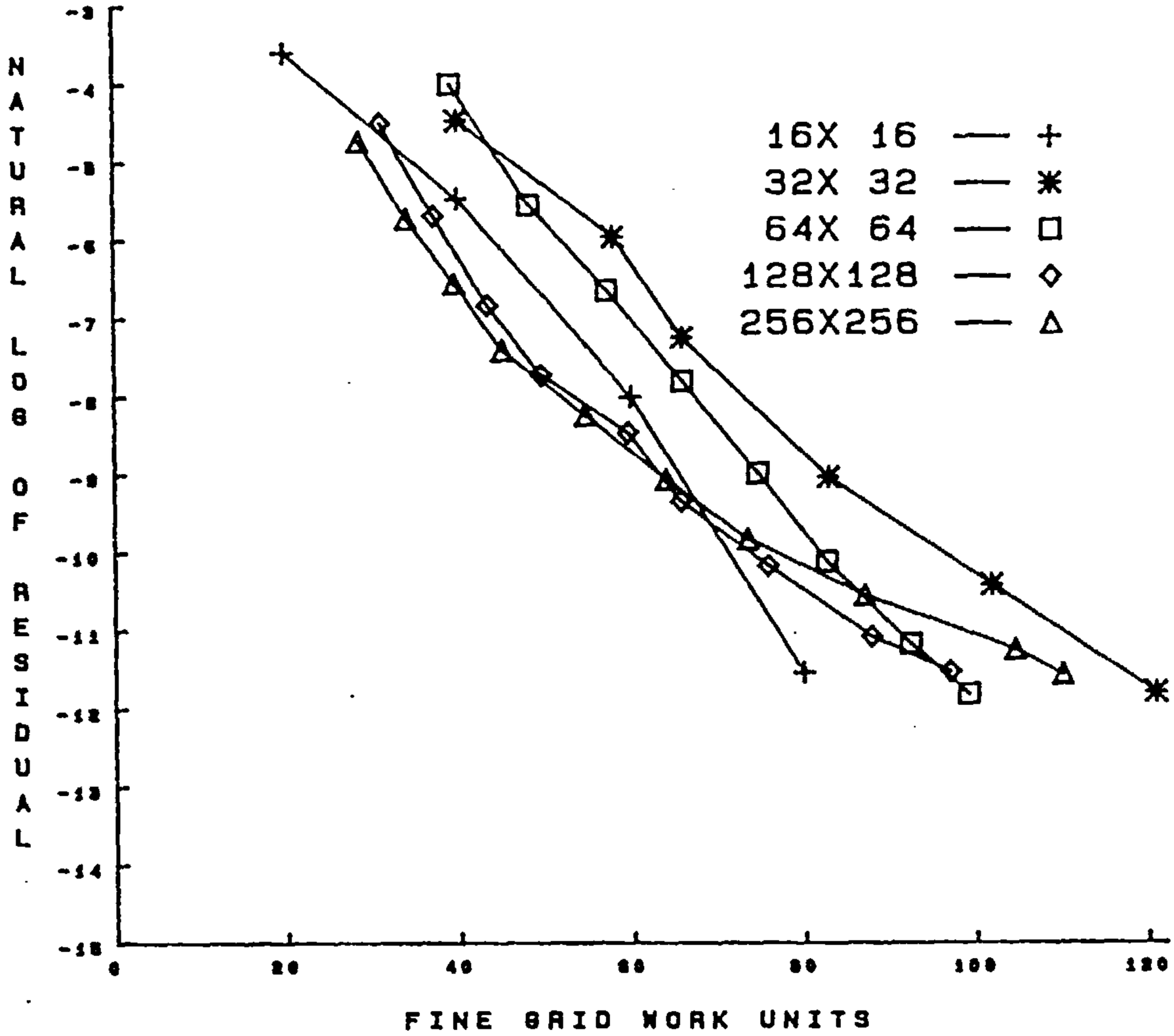


Figure 5.15: Plot of natural log of residual against FGWU for Reynolds number 50 obtained with CCCT ($\alpha = 0$) differencing.

hypotheses leads to an infinite vorticity at this point, which the numerical solution tries to reflect (this can be observed from Figure 5.14). The nature of this problem as posed numerically is different on each level. This inconsistency leads to degradation of convergence[8].

(iii) The smoother is unsuited to such an unidirectional flow with sharp gradients present in certain areas, as observed by Linden et al.[7]. The smoother itself does not perform well, and so effects the multigrid convergence.

(iv) The physics of the flow itself is more complex than the geometry might suggest. Fuchs[3], for example, observed that even for moderate Reynolds number the flow exhibits oscillatory solutions and bifurcations.

The unsatisfactory convergence is highlighted in Table 5.7 of multigrid converence rates (defined in equation (4.16)). The values are much higher than those for both the two- and three-dimensional lid-driven cavity, but even so they are less than they would be for a non-multigrid techique, where they would be greater than 0.99 on finer meshes.

There are several possible ways of overcoming these problems which are discussed in Chapter 7.

References

1. , "Analysis of Laminar Flow over a Backward Facing Step", in *Notes on Numerical Fluid Dynamics*, ed. Morgan, Periaux and Thomasset, (1984).

2. Falle, S.A.E.G. and Wilson, M.J., "Private Communication", *University of Leeds*, (1986).

3. Fuchs, L., "An Adaptive Multigrid Scheme for the Simulation of Flows", pp. Springer-Verlag in *Multigrids Methods II*, ed. Hackbusch and Trottenberg, (1985).

4. Gaskell, P.H. and Lau, A.K.C., "An Efficient Solution Strategy for Use with Higher Order Discretisation Schemes", *Report No. T40, Department of*

Mechanical Engineering, University of Leeds, (1986).

5. Gaskell, P.H. and Lau, A.K.C., "Curvature Compensated Convective Transport : SMART, a New Boundedness Preserving Transport Algorithm", *International Journal for Numerical Methods in Fluids* **8** pp. 617-641 (1988).

6. Leonard, B.P., "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", *Computational Methods in Applied Mechanics and Engineering* **19** pp. 59-98 (1979).

7. Linden, J., Steckel, B., and Stuben, K., "Parallel Multigrid Solution of the Navier-Stokes Equation on General 2D-Domains", *Arbeitspapiere der GMD*, (294)(1988).

8. Rude, U. and Zenger, Chr., "On the Treatment of Singularities in the Multigrid Method", pp. Springer-Verlag in *Multigrids Methods II*, ed. Hackbusch and Trottenberg, (1985).

9. Sivaloganathan, S., "Private Communication", *Oxford University Computing Laboratory*, (1987).

10. Spalding, D.B., "A Novel Finite Difference Formulation for Differential Expressions Involving both First and Second Derivatives", *International Journal for Numerical Methods in Engineering* **4** pp. 551-559 (1972).

11. Vanka, S.P., "Block-Implicit Multigrid Calculation of Three-Dimensional Recirculating Flows", pp. 667-679 in *Proceedings of the 4th International Conference on Numerical Methods for Thermal Problems, Part 1*, , Swansea (1985).

12. Vanka, S.P., "Performance of a Multigrid Calculation Procedure in Three-Dimensional Sudden Expansion Flows", *International Journal for Numerical Methods in Fluids* **6** pp. 459-477 (1986).

Chapter 6

THERMALLY DRIVEN FLOWS

6.1. Introduction

Work reported so far has focussed on fluid flow problems of a complex nature, but ones that involve only the transferral of mass and momentum in a conservative system. However, in an engineering environment, many problems of practical interest may also involve additional physical properties such as turbulence or energy production via combustion, or some other means. In which case the governing equations of motion multiply in number, introducing additional parameters into the system. Could a multigrid solution strategy be adapted here?

Processes invoving heat release, such as chemical reaction, can be extremely difficult to model and present computational fluid dynamicists with a rather acute problem, due to the inclusion of variable density. Consequently, the treatment of such flows in best left to a competent practitioner. Little work has been reported in this area, and therefore it is difficult to comment here on the benefits that might be accrued from the use of a multigrid approach in solving such problems.

In the case of turbulence the additional parameters that arise, derive from the necessary use of turbulence models[7] in order to make such problems tractable. They are, needless to say,inherently complex and computationally demanding - even for the simplest of flows. One major drawback of such models is the use of wall functions as a means of predicting the flow behaviour at nodal points adjacent to solid boundaries. Traditionally this has proved to be a most satisfactory way of accounting for laminar boundary layers that cannot be adequately resolved since, in general, their physical length-scale is less than the computational mesh spacing. However, it is envisaged that this would introduce incompatibilities between meshes when using a multigrid technique. Nevertheless, it is not premature to postulate that it should eventually be possible to solve such systems with the aid of multigrids.

Fortunately, a class of problems does exist that are of great practical interest and yet are not problematic in the sense of the flows described above. These are thermal problems which occur in abundance in several ares such as power generation, domestic

heating and insulation, climatic and environmental change. The equations of motion are a simple extension of those given in Chapter 3

$$\frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j^2} + \delta_{i2} g\beta T \tag{6.1}$$

$$\frac{\partial}{\partial x_j}(u_j T) = \frac{k}{c_p \rho}\frac{\partial^2 T}{\partial x_j^2} \tag{6.2}$$

These can be non-dimensionalised using

$$u_i' = \frac{u_i}{u_r}, \quad p' = \frac{p}{\rho u_r^2},$$

$$x_i' = \frac{x_i}{L}, \quad \theta = \frac{T}{\Delta T},$$

where $u_r = g\beta\Delta T L^{1/2}$, a reference velocity

L is the width of the cavity

ΔT is the temperature difference.

Ignoring the dashed superscript this gives

$$\frac{\partial}{\partial x_j}(u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{1}{Gr^{1/2}}\frac{\partial^2 u_i}{\partial x_j^2} + \delta_{i2}\theta \tag{6.3}$$

$$\frac{\partial}{\partial x_j}(u_j \theta) = \frac{1}{Gr^{1/2} Pr}\frac{\partial^2 \theta}{\partial x_j^2} \tag{6.4}$$

where

$$\text{Prandtl number, } Pr = \frac{c_p \mu}{k}$$

$$\text{Grashof number, } Gr = \frac{g\beta\Delta T L^3}{\nu^2}.$$

and

$$\text{Rayleighnumber, } Ra = Gr\ Pr$$

These equations make use of the Boussinesq approximation for steady flows, which is valid for small values of θ. This assumes that density variations are negligible except in the buoyancy term in the momentum equation It involves assuming that ρ becomes $\rho_o\beta\Delta T$ in the buoyancy term and ρ_o elsewhere, where ρ_o is a constant.

The temperature equation (6.4) written in two-dimensions

$$\frac{\partial u\theta}{\partial x} + \frac{\partial v\theta}{\partial y} = \frac{1}{Gr^{½}\,Pr}\left[\frac{\partial^2\theta}{\partial x^2} + \frac{\partial^2\theta}{\partial y^2}\right]. \tag{6.5}$$

is discretised as follows. The temperature node is situated at the centre of the finite control volume, in the same place as the pressure. Equation (6.5) can then be integrated over a control volume to give

$$\frac{(u\theta)_{i-½j}-(u\theta)_{i+½j}}{h} + \frac{(v\theta)_{ij+½}-(v\theta)_{ij-½}}{h} = \frac{1}{Gr^{½}Pr}\left[\frac{\theta_{i+1j}-\theta_{ij}}{h^2}-\frac{\theta_{ij}-\theta_{i-1j}}{h^2} + \frac{\theta_{ij+1}-\theta_{ij}}{h^2} - \frac{\theta_{ij}-\theta_{ij-1}}{h^2}\right] \tag{6.6}$$

or

$$\frac{u_{ij}\theta_{i+½j} - u_{i-1j}\theta_{i-½j}}{h} + \frac{v_{ij}\theta_{ij+½}-v_{ij-1}\theta_{ij-½}}{h} = \frac{1}{Gr^{½}Pr}\left[\frac{\theta_{i+1j}+\theta_{i-1j}+\theta_{ij+1}+\theta_{ij-1}-4\theta_{ij}}{h^2}\right]. \tag{6.7}$$

The temperature at faces of the control volume are calculated using interpolation, the velocities there, on the other hand, are known calculated by using interpolation. The value of T required in the vertical momentum equation is calculated using linear interpolation.

The Block Implicit Method is easily adapted to solve for thermal flows. In the two-dimensional case a sixth equation is obtained in addition to the continuity and four momentum equations already encountered. To include this equation in the matrix given in equation (3.32) would destroy its doubly bordered form and therefore necessitate the use of a less efficient solver. Instead, the coefficient matrix (3.32) is solved as before, the additional equation for temperature being solved seperately afterwards. This leads to some decoupling of the thermal effects. However, to include these effects efficiently would require the use of a different solver. Galpin and Raithby[4] have discussed the importance of temperature-velocity coupling and

proposed a scheme that satisfactorily accounts for the former. It is based on the CELS technique (see sub-section 3.4.3), but for the reasons outlined in earlier discussions this method was not considered to represent a viable alternative for use in evaluating a multigrid approach to recirculating thermal flows. It was also decided to stick with the solver used so far which gave good results for the non-thermal problems, and is superior to SIMPLE (which is troublesome for thermal problems, as outlined by Raithby[4] and experienced by work at Rolls-Royce.)[9] A further relaxation factor has to be introduced for the temperature. The residual used is analogous to that in equation (3.33). i.e.

$$|| r || = \frac{\left(\sum_{i,j} \left(r_u^{ij^2} + r_v^{ij^2} + r_c^{ij^2} + r_\theta^{ij^2} \right) \right)^{1/2}}{4 \times n \times n}. \qquad (6.8)$$

The multigrid technique needs very little modification for thermal flows. One extra equation is added for each grid. The temperature is restricted and prolonged in the same manner as the pressure, apart, that is, from adaptations at the boundaries, which are outlined below for each of the test problems investigated. In view of the comparative performance of hybrid and CCCT(α=0)[5] differencing it was decided to make use of the latter only.

6.2. A Thermally Driven Cavity with Conducting Walls

In addition to equation (6.2) this problem is defined by the following boundary conditions, see Figure 6.1(a),

(a) on x=0, u=v=0, θ=1

(b) on x=1, u=v=0, θ=0

(c) on y=0 and y=1, u=v=0, θ =1-x.

This corresponds to a thermally driven cavity with conducting top and bottom walls. Solutions were found for this problem with Prandtl number, Pr = 0.71, (corresponding to air) and Rayleigh numbers of 10^3, 10^4, 10^5, and 10^6, on meshes containing upto 256^2 internal nodes. Rayleigh numbers greater than these have been observed, both numerically and experimentally, to give bifurcating solutions[1] and so were not sought

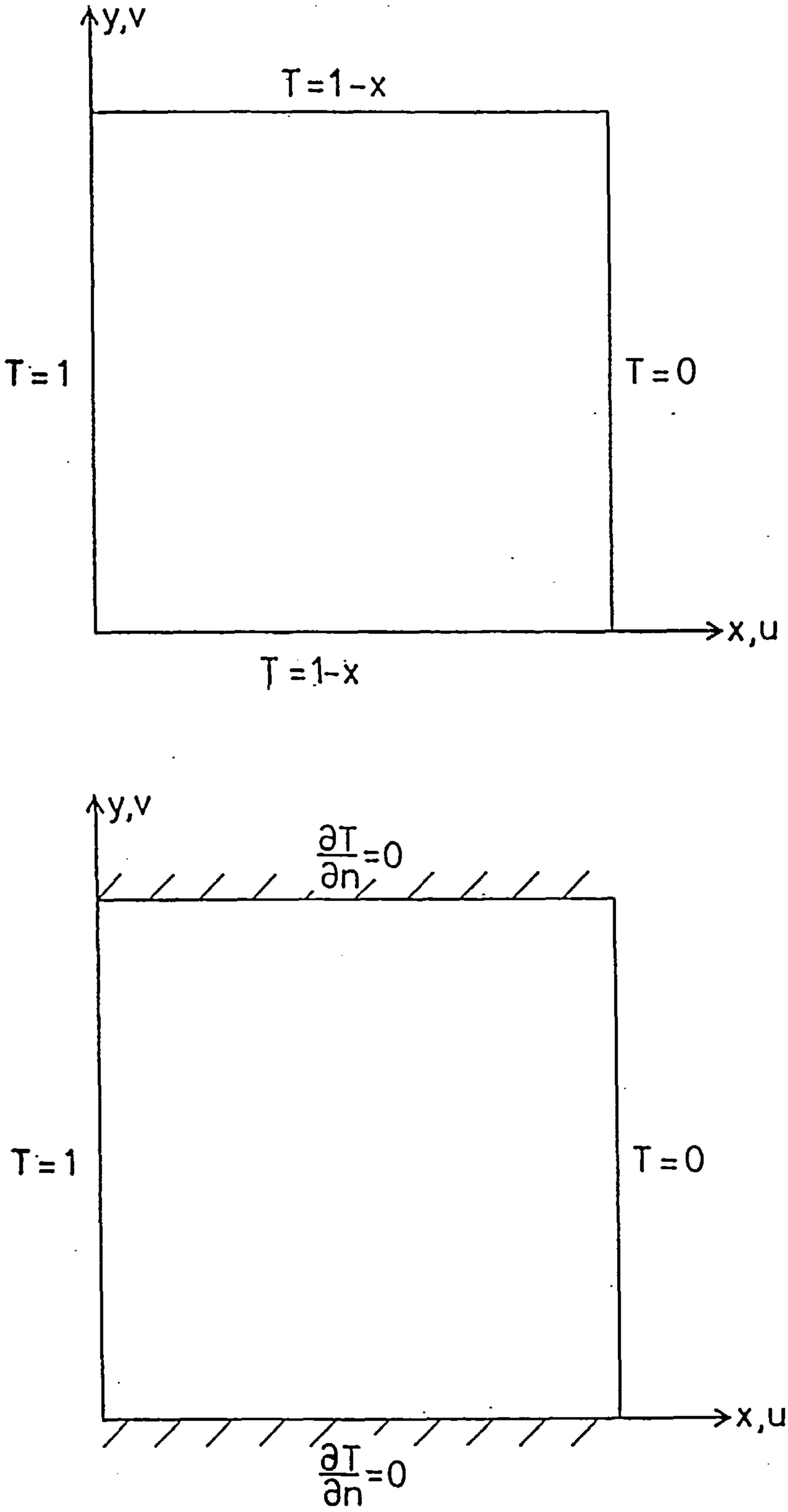


Figure 6.1: Flow configuration and boundary conditions for the thermally driven cavity with (a) conducting and (b) insulated top and bottom walls

here.

Prolongation of near boundary nodes was accomplished in the same way as in the case of the two-dimensional lid-driven cavity, that is a zero derivative condition was employed there. The results for various numbers of nodes and with CCCT(α=0) discretisation are shown in Tables 6.1 and 6.2 and in Figures 6.2 to 6.5. The ψ_{mid} is the value of the streamfunction at (x,y) = (0.5,0.5). It is calculated as the mean of the surrounding four values. ψ_{max} is the maximum nodal value of ψ in the flow. again x and y give the position where the maximum occurs. $\overline{Nu}$ is the average Nusselt number over the cavity. This is defined as

$$\overline{Nu} = \frac{1}{V}\int_V Q(x,y)dV, \tag{6.9}$$

where

$$Q(x,y) = u\theta - \frac{\partial\theta}{\partial x}$$

Nu_{max} is the maximum Nusselt number in the vertical plane and y its position and similarly for Nu_{min} and x. U_{max} is the maximum horizontal velocity on the vertical mid plane at y. Similarly, V_{max} is the maximum vertical velocity on the horizontal mid plane at x.

Whilst at $Ra=10^3$ the temperature contours (see Figure 6.2(c)) remain roughly vertical they begin to bend at $Ra=10^4$ (Figure 6.3(c)) and at $Ra=10^5$ (Figure 6.4(c)) a central zone of horizontal contours is seen to have developed. This results in a region of flow that is slowly moving with respect to the bulk of the fluid, which can be seen in Figure 6.4(a). This phenomenon can also be observed in Figures 6.5(a) and (b) depicting streamfunction and vorticity, particularly when $Ra=10^6$. As this region develops the position of the maximum streamfunction (see Table 6.1) leaves the centre of the cavity and two equal maxima develop. The regions of greatest vorticity are adjacent to the boundary (the two vertical ones). These regions present most difficulty for both discretisation and solution technique. Such widely varying regimes point to the use of adaptive meshing[8], which is discussed in this context in Chapter 7.

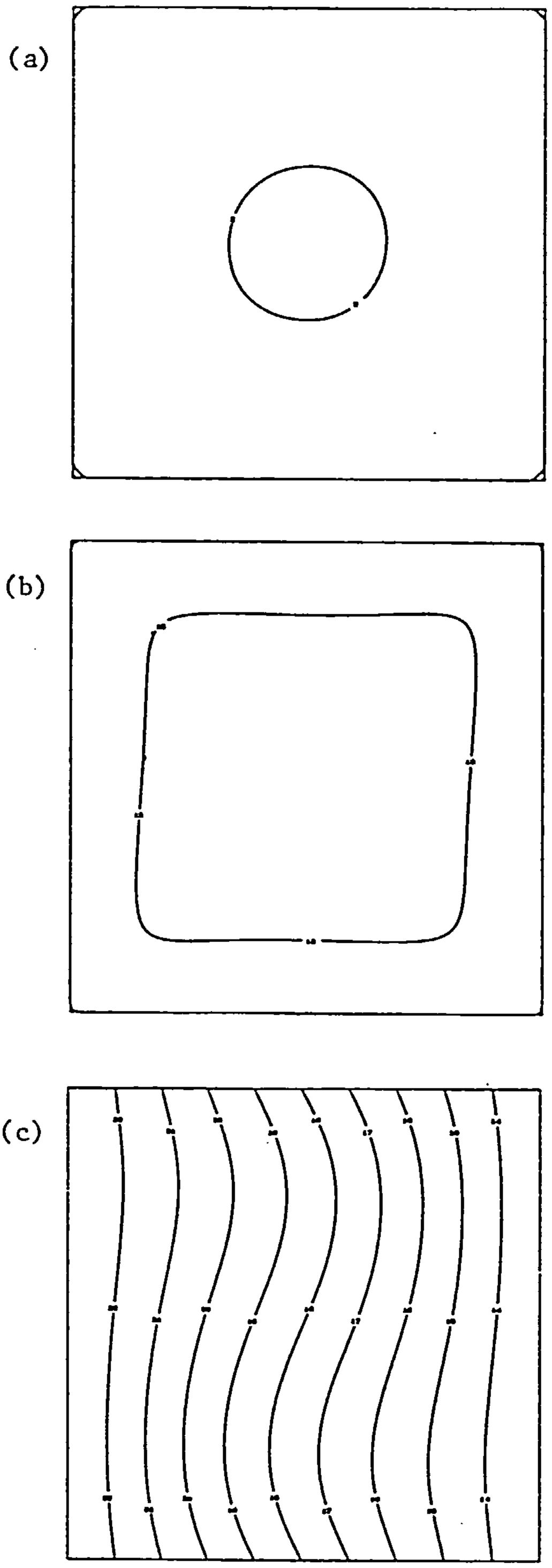


Figure 6.2: Contour plots for Rayleigh number 10^3 with 256^2 internal nodes, for the thermally driven cavity with conducting top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

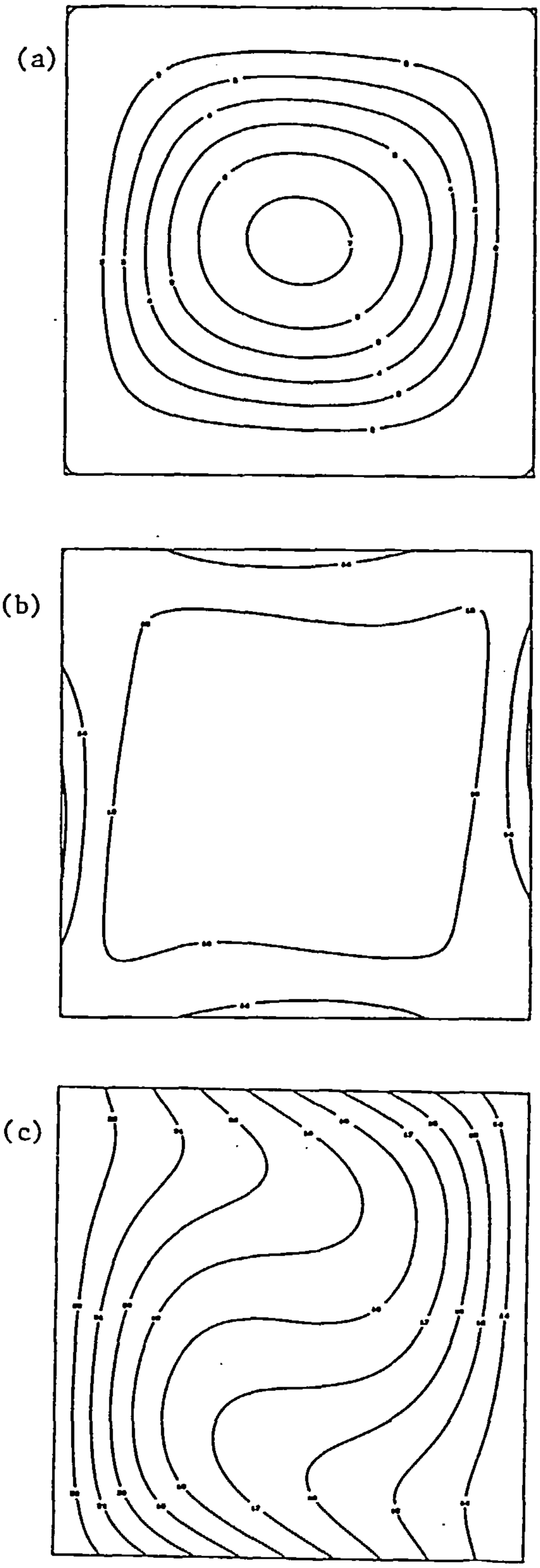
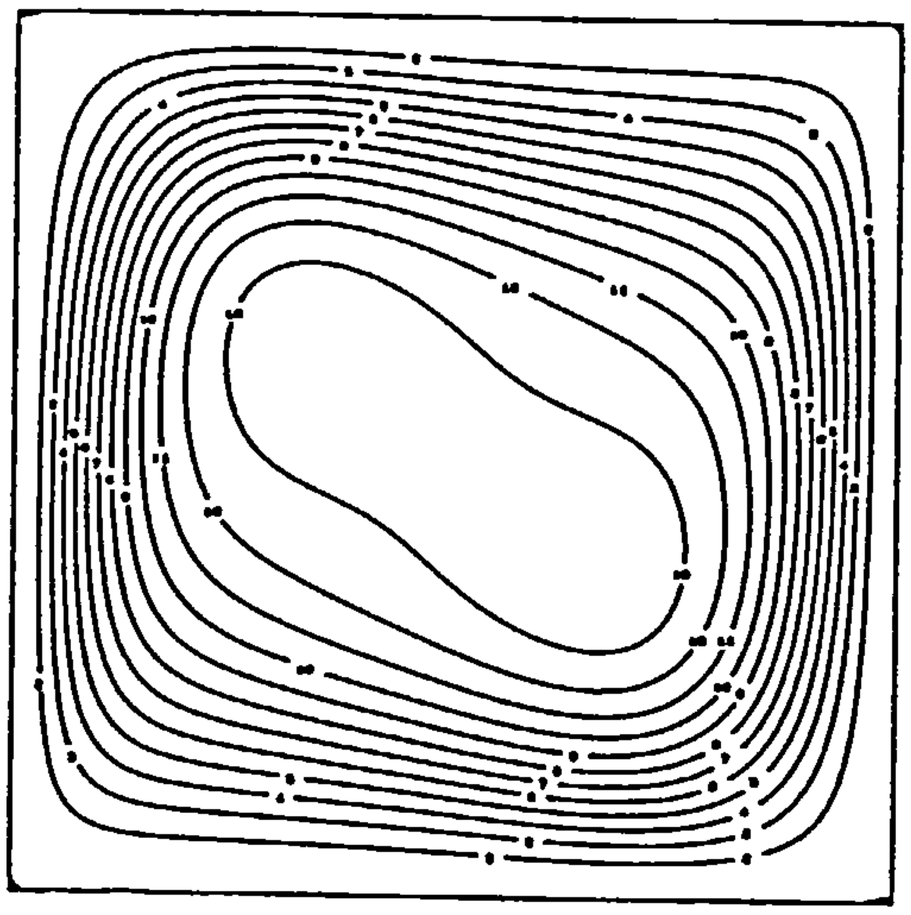
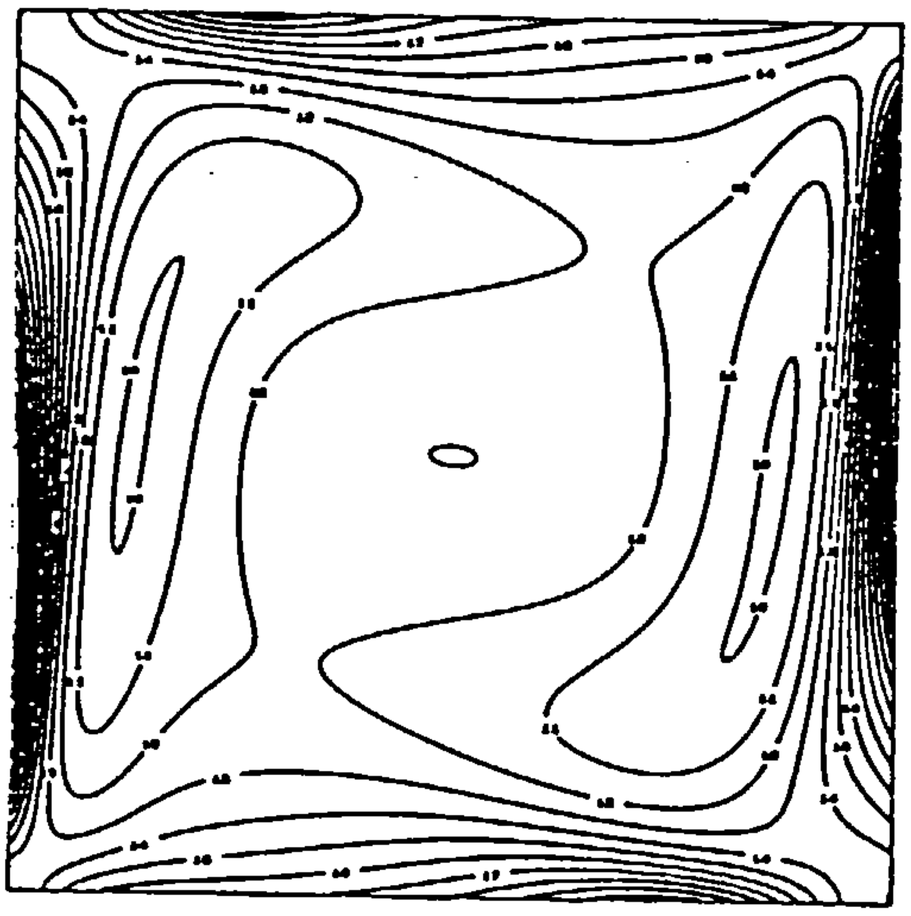


Figure 6.3: Contour plots for Rayleigh number 10^4 with 256^2 internal nodes, for the thermally driven cavity with conducting top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

(a)



(b)



(c)

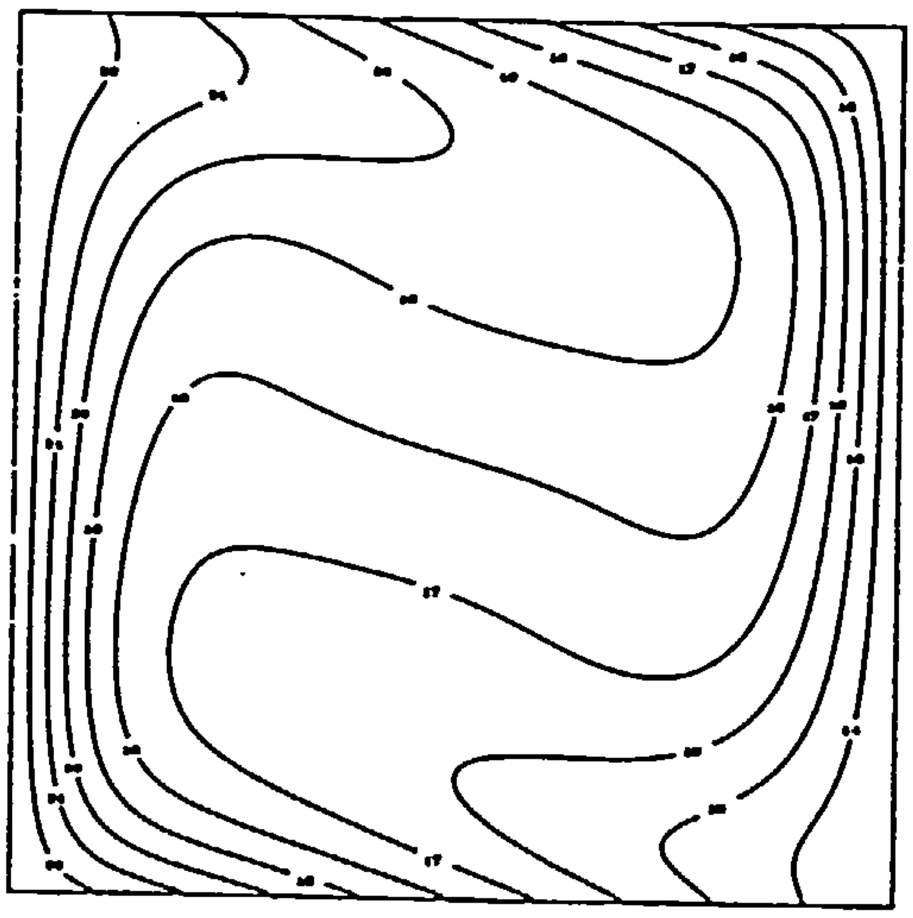


Figure 6.4: Contour plots for Rayleigh number 10^5 with 256^2 internal nodes, for the thermally driven cavity with conducting top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

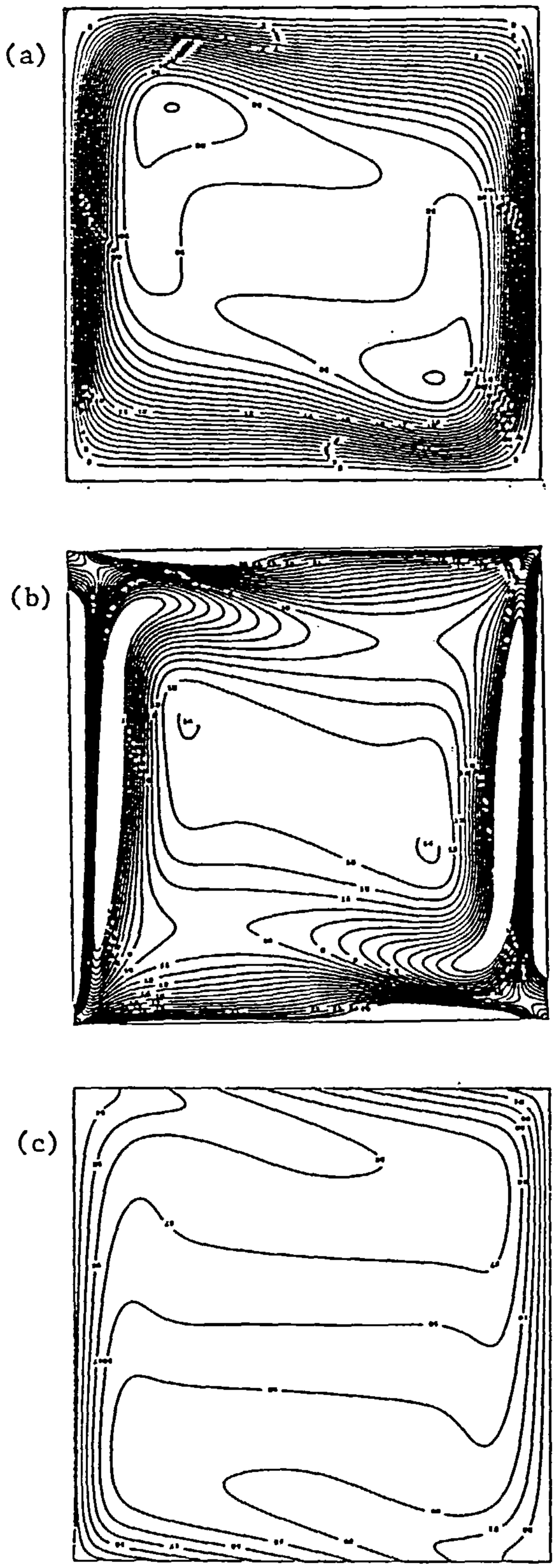


Figure 6.5: Contour plots for Rayleigh number 10^6 with 256^2 internal nodes, for the thermally driven cavity with conducting top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

Ra	10^3	10^4	10^5	10^6
Ψ_{mid}	1.220	6.338	12.260	19.6115
Ψ_{max}	-	-	12.662	22.047
x	-	-	0.336	0.2195
y	-	-	0.625	0.784
U_{max}	3.796	20.377	55.21	124.20
y	0.815	0.834	0.881	0.904
V_{max}	3.812	22.999	80.777	253.60
x	0.178	0.127	0.068	0.041
$\overline{Nu}$	1.058	1.920	3.785	7.2992
Nu_{max}	1.225	2.521	4.942	23.283
y	0.258	0.305	0.289	0.212
Nu_{min}	0.871	0.838	0.965	2.324
y	0.793	0.891	0.965	1.0

Table 6.1: Selected characteristic data for the thermally driven cavity with conducting walls.

	Rayleigh number			
Mesh	10^3	10^4	10^5	10^6
4^2	0.22	0.34	0.60	-
8^2	1.23	1.87	3.98	-
16^2	2.47	3.95	7.94	27.57
32^2	7.10	8.87	17.10	81.61
64^2	25.01	27.74	38.97	139.39
128^2	96.11	99.68	114.68	280.25
256^2	377.87	382.23	399.65	700.67

Table 6.2: Computer time for the solution of the thermally driven cavity with conducting walls.

The computer times for this flow are given in Table 6.2. These increase for increasing Rayleigh number, because of the increasing complexity of the flow, particularly in the near boundary regions. The requirement that cpu time is proportional to the number of nodes is more than satisfied. As the number of nodes increases the FGWU requirement decreases, and the use of a large number of levels leads to optimal multigridding. Figures 6.6 to 6.9 show the log of residual against FGWU. As seen in Chapter 4 this should be a straight line, which from examination of these graphs is seen to be the case. It can also be observed, again, that multigridding becomes more optimal with an increase in the number of levels. Table 5.3 gives the multigrid convergence rate as defined in equation (4.16). They are less than those for the two-dimensional cavity, but direct comparison is not valid because of the different

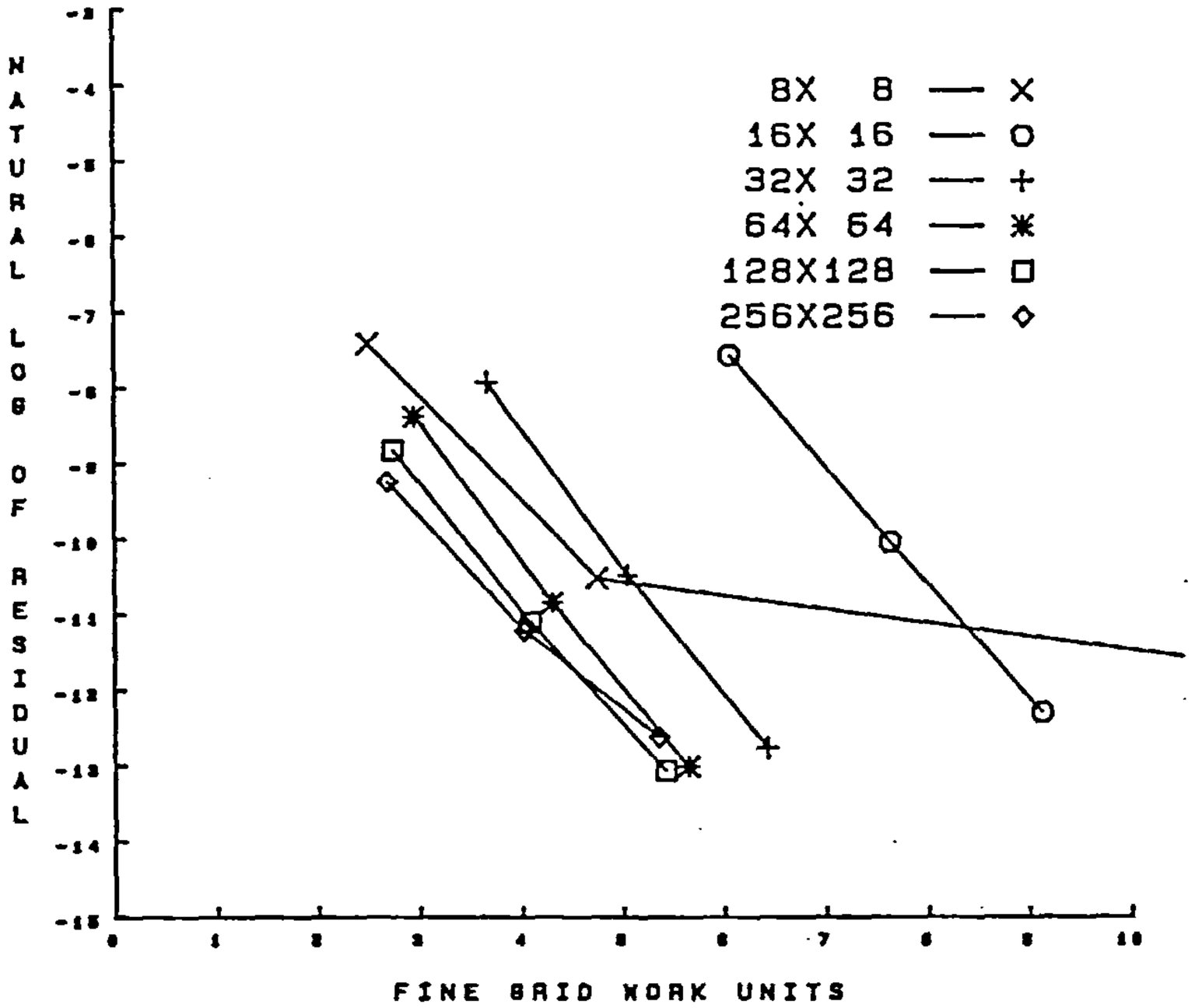


Figure 6.6: Plot of the natural log of the residual against FGWU for Rayleigh number 10^3 in the case of the thermally driven cavity with conducting top and bottom walls.

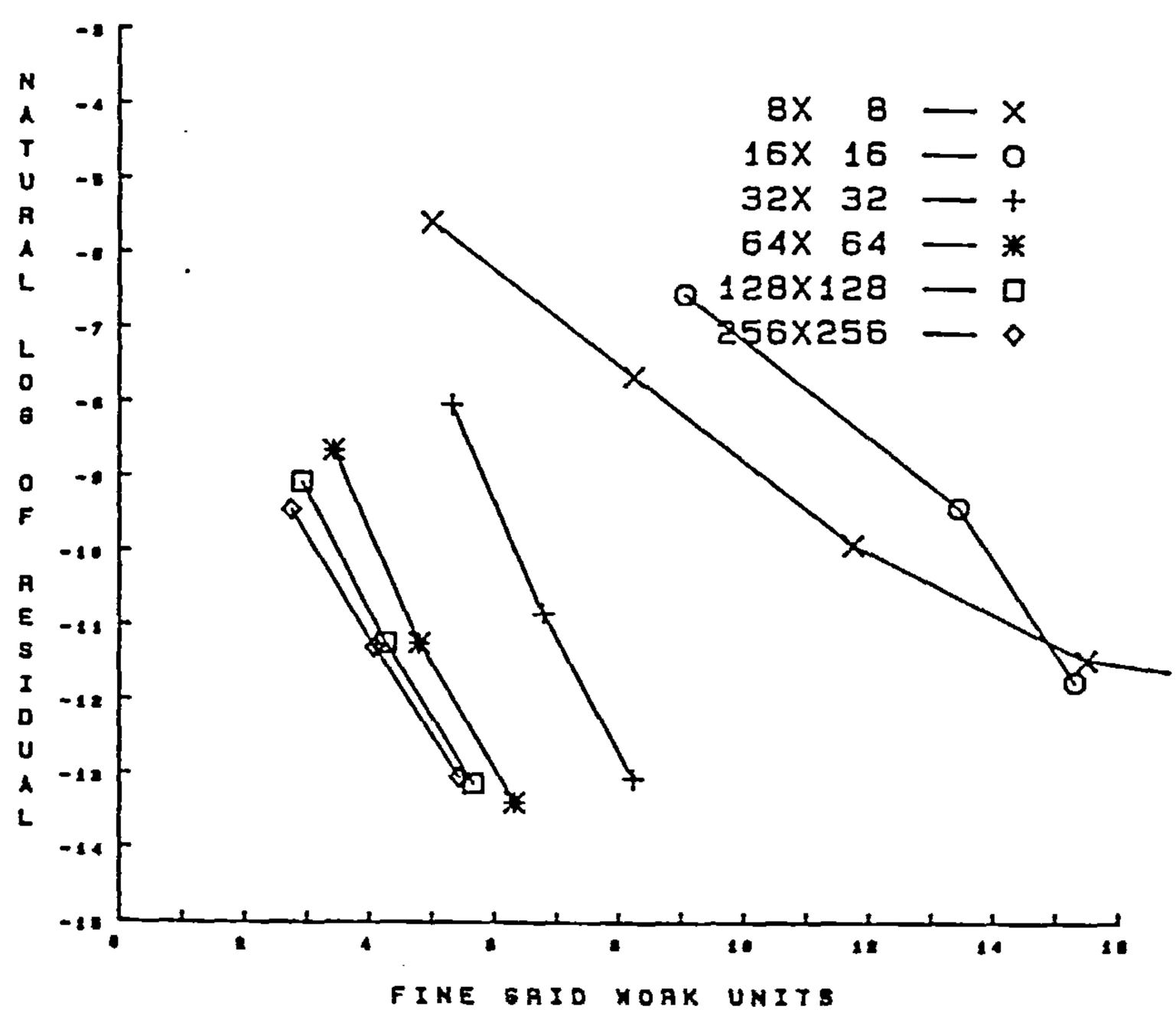


Figure 6.7: Plot of the natural log of the residual against FGWU for Rayleigh number 10^4 in the case of the thermally driven cavity with conducting top and bottom walls.

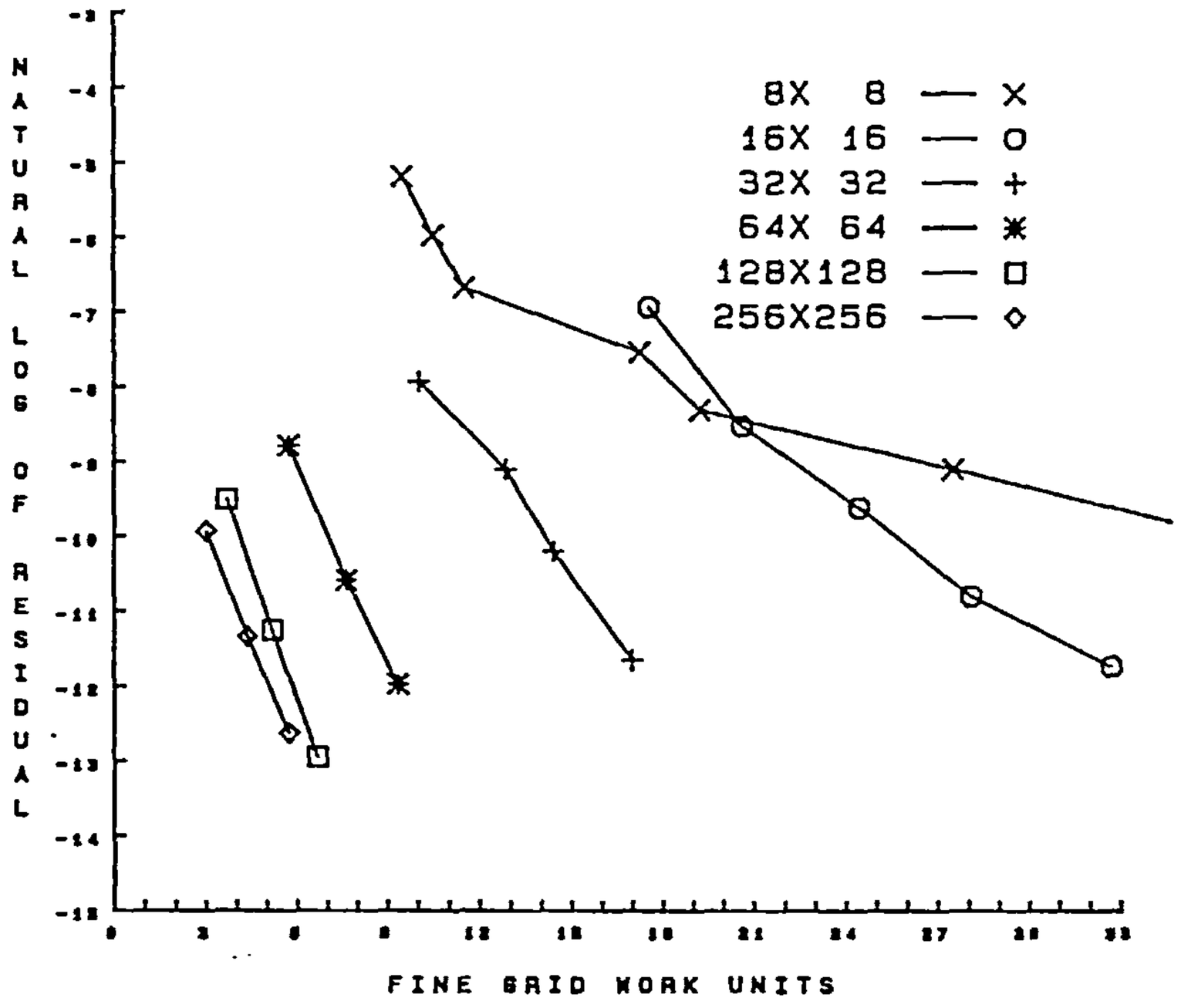


Figure 6.8: Plot of the natural log of the residual against FGWU for Rayleigh number 10^5 in the case of the thermally driven cavity with conducting top and bottom walls.

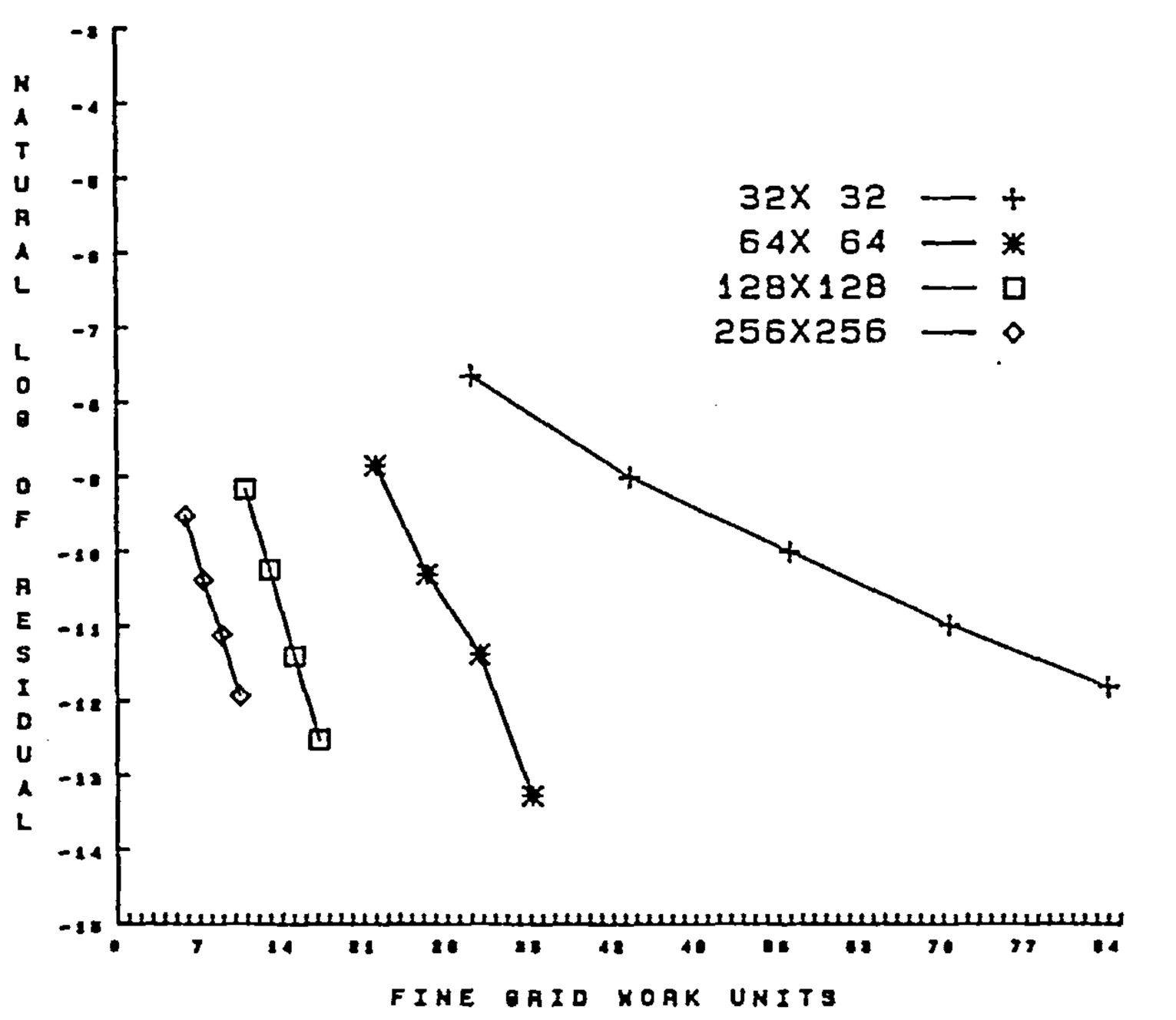


Figure 6.9: Plot of the natural log of the residual against FGWU for Rayleigh number 10^6 in the case of the thermally driven cavity with conducting top and bottom walls.

Ra	10^3	10^4	10^5	10^6
4^2	0.40	0.48	0.65	-
8^2	0.62	0.69	0.88	-
16^2	0.22	0.47	0.66	-
32^2	0.20	0.23	0.56	0.90
64^2	0.20	0.22	0.40	0.64
128^2	0.21	0.24	0.31	0.57
256^2	0.26	0.26	0.33	0.54

Table 6.3: Multigrid convergence rates for the cases considered in section 6.2

flow regime and different dimensionless parameters. However, the values indicate that incorporating thermal effects does not a detrimental effect on multigrid convergence.

Boonkkamp solved this problem using a time dependent set of equations to converge to a steady state. His method used an alternating direction Euler method with a pressure correction. The results, while at different Rayleigh numbers, bear out the solutions obtained here. Boonkkamp's main investigation was of the bifurcation of solutions to the flow at Rayleigh number of 3.0×10^6.

6.3. A Thermally Driven Cavity with Insulating Walls

This problem is a more widely known extension of the previous one . It is often referred to as the "Double Glazing" problem, for obvious reasons. It differs from the above case in that the the top and bottom walls are perfectly insulated, (see Figure 6.1(b)) and so the boundary conditions become,

(a) on all faces $u=v=0$

(b) $\theta=1$ on $x=0$

(c) $\theta=0$ on $x=1$

(d) $\frac{\partial \theta}{\partial n} = 0$ on $y=0$ and $y=1$.

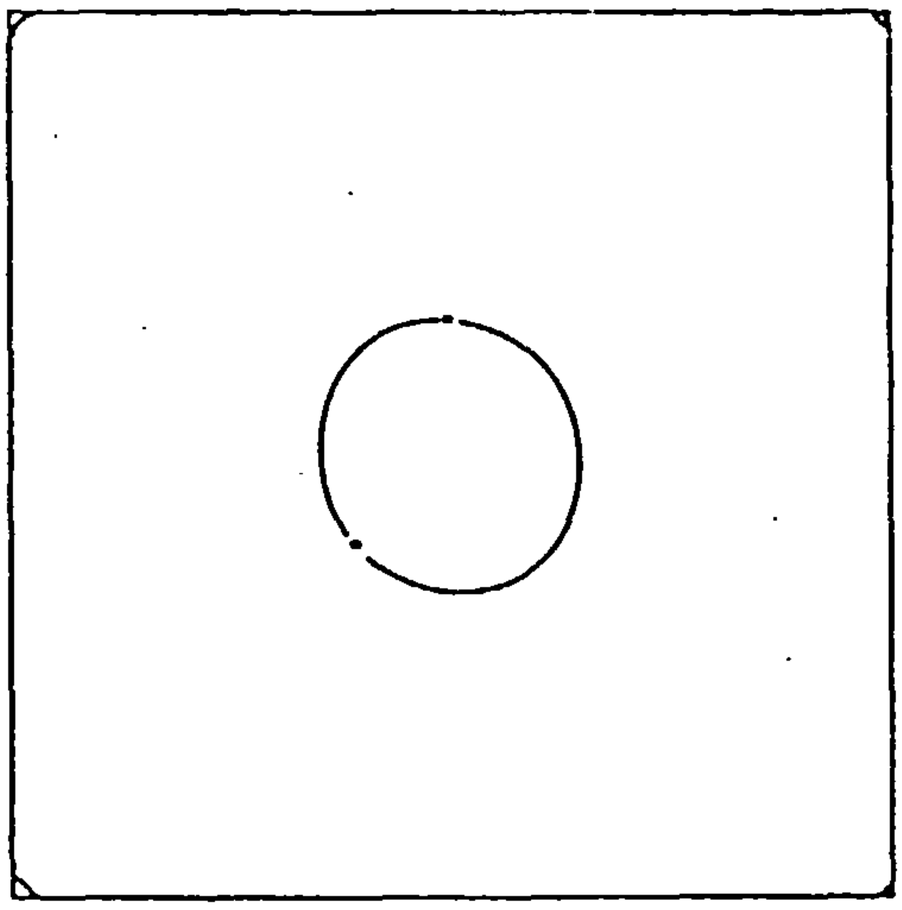
The derivative boundary condition is dealt with as described in Chapter 5. The near boundary values of the temperature are prolonged using the boundary values. The boundary values themselves are prolonged as well.

The results are presented in Tables 6.4, 6.5 and 6.6 and in Figures 6.10 to 6.13. Solutions were again obtained for Rayleigh numbers of 10^3, 10^4, 10^5 & 10^6 at Prandtl

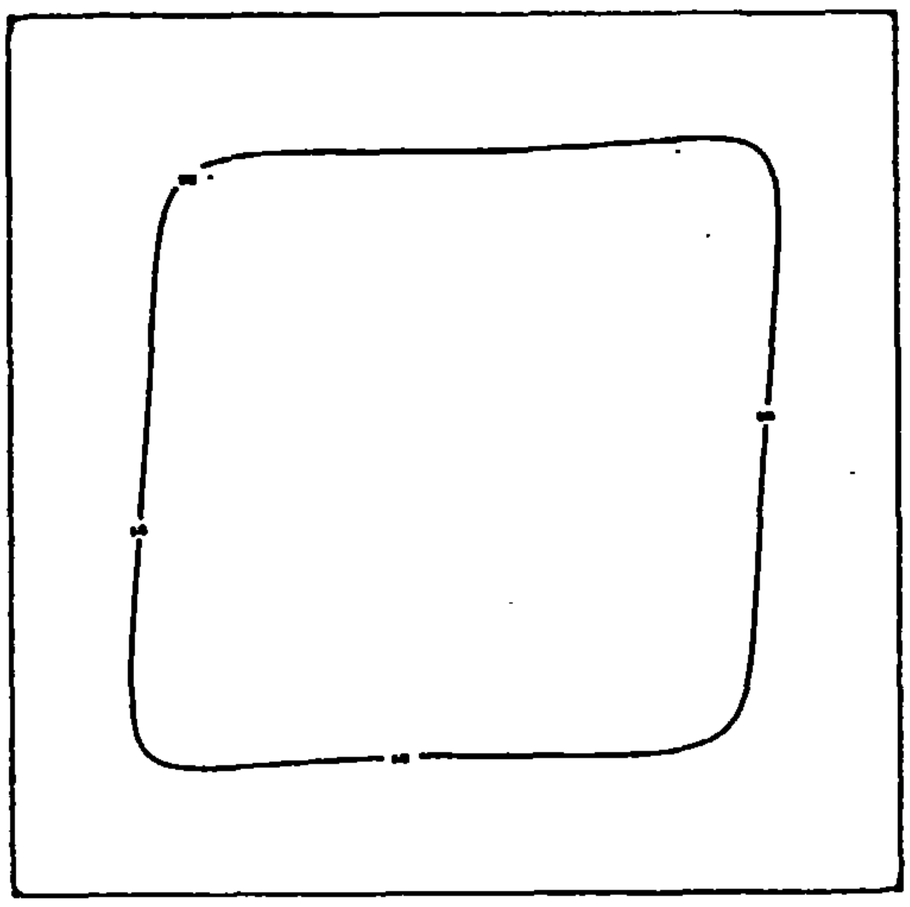
Ra	10^3	10^4	10^5	10^6
Ψ_{mid}	1.178	5.068	9.083	16.444
Ψ_{max}	-	-	9.585	16.743
x	-	-	0.281	0.152
y	-	-	0.598	0.547
U_{max}	3.660	16.135	34.17	64.14
y	0.815	0.822	0.854	0.850
V_{max}	3.697	19.622	68.44	219.33
x	0.178	0.119	0.065	0.037
$\overline{Nu}$	1.117	2.251	4.564	8.870
Nu_{max}	1.498	3.460	7.797	18.078
y	0.125	0.156	0.090	0.039
Nu_{min}	0.650	0.572	0.751	0.980
y	1.0	1.0	1.0	1.0

Table 6.4: Selected characteristic data for the thermally driven cavity with adiabatic walls.

(a)



(b)



(c)

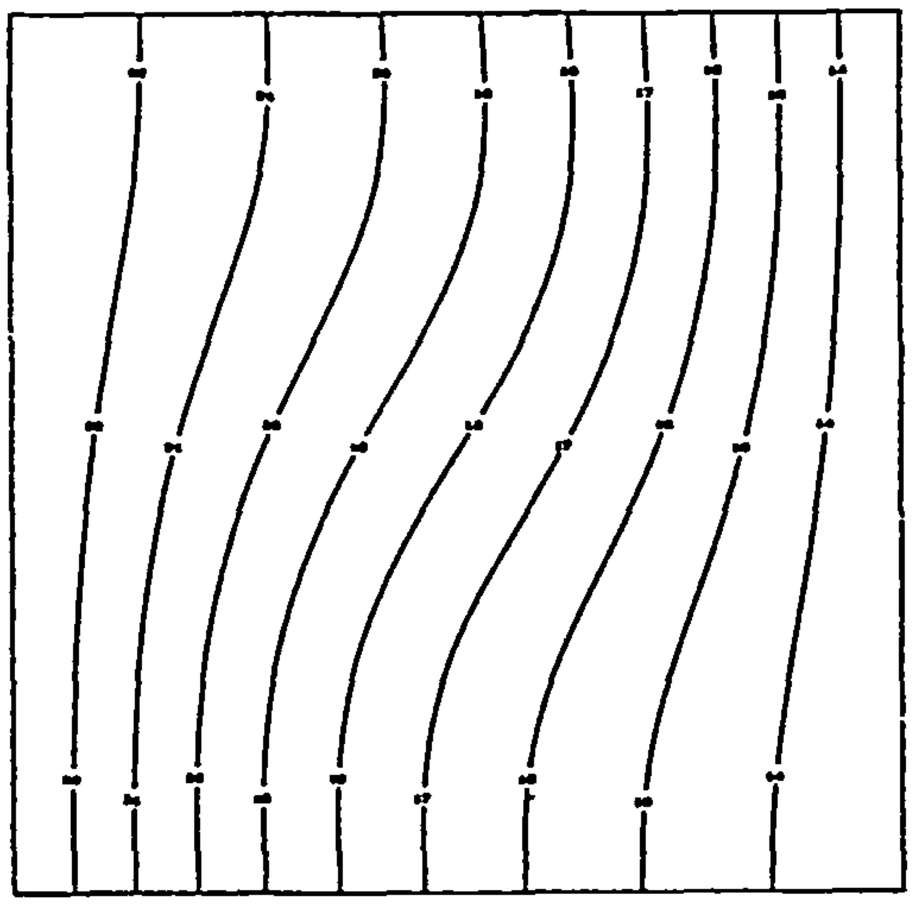
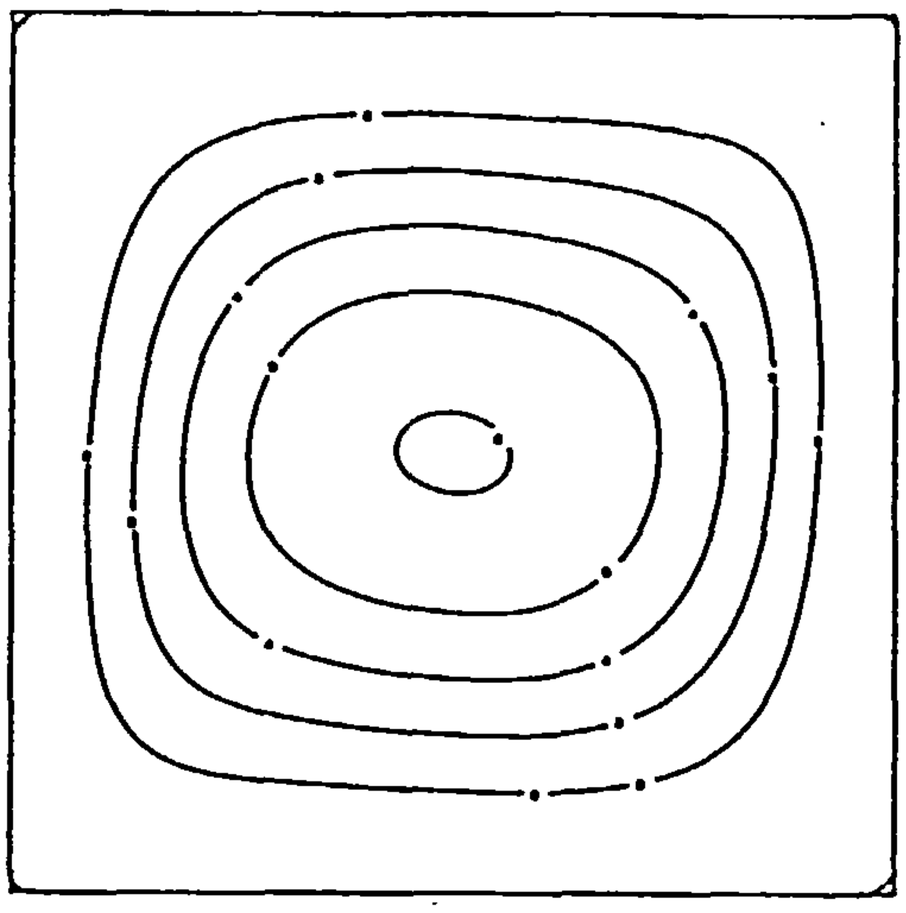
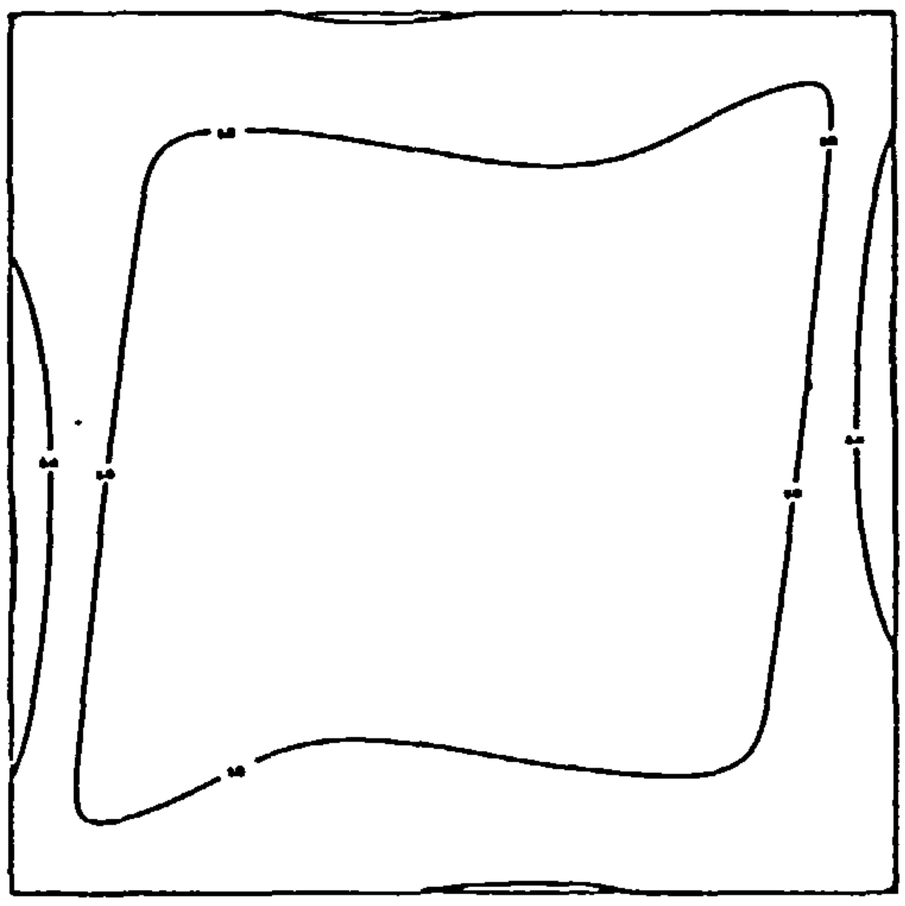


Figure 6.10: Contour plots for Rayleigh number 10^3 with 256^2 internal nodes, for the thermally driven cavity with insulated top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

(a)



(b)



(c)

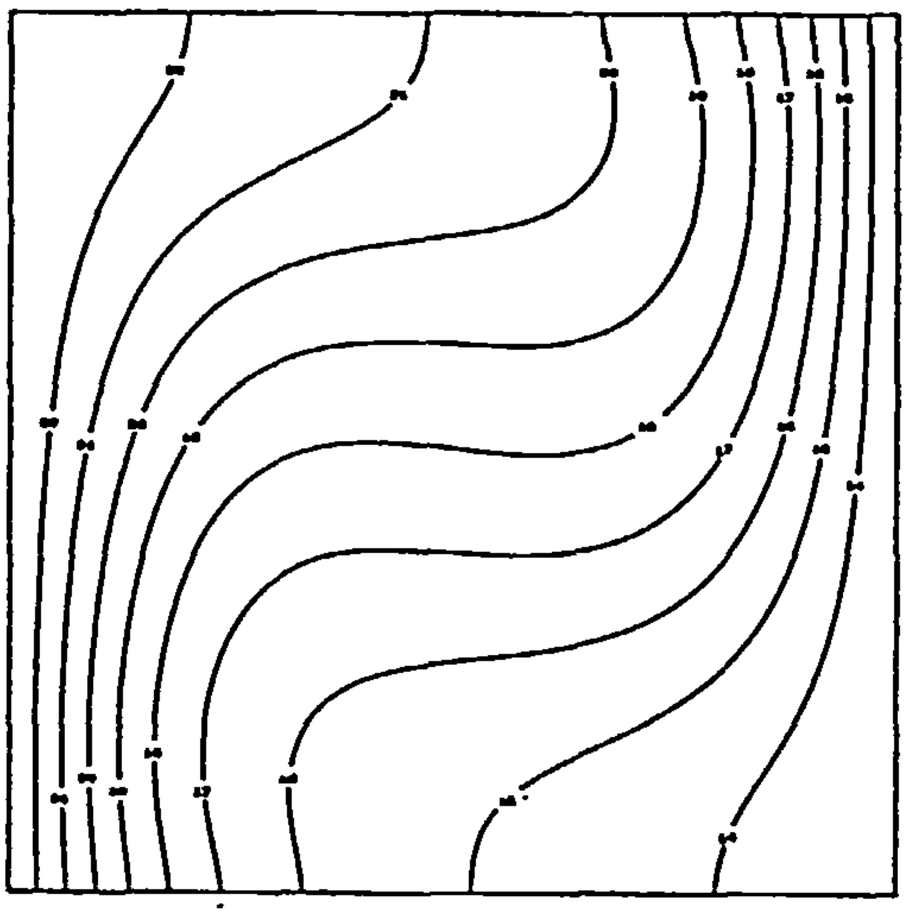
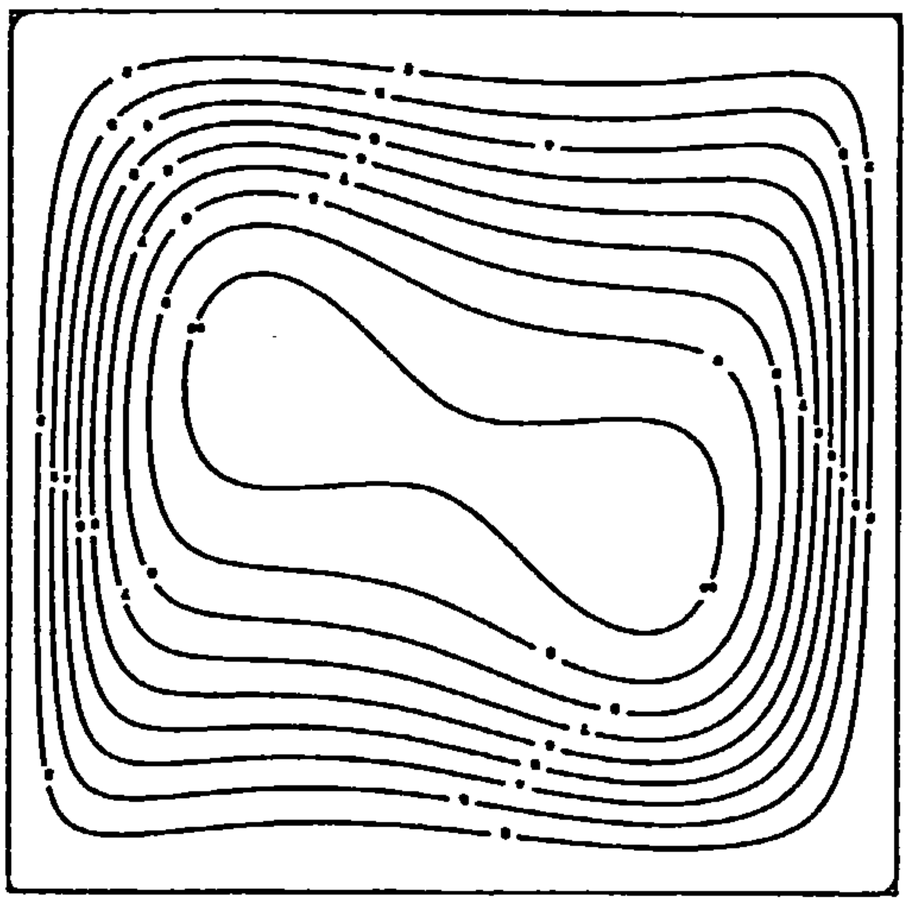
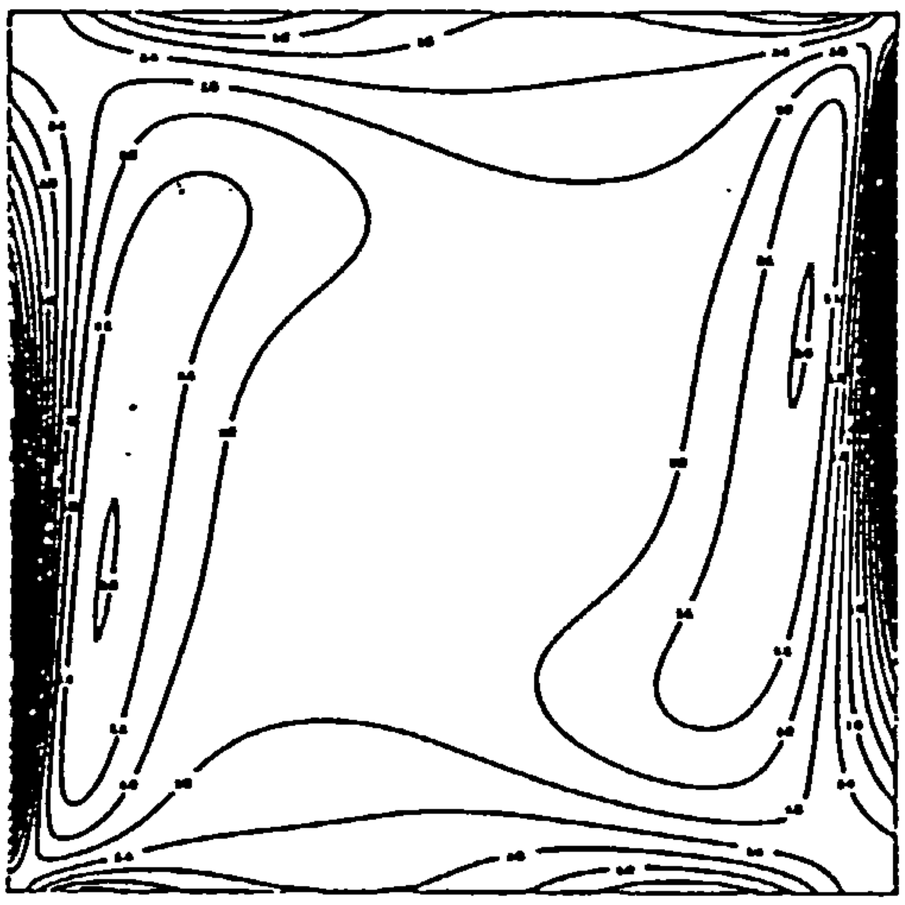


Figure 6.11: Contour plots for Rayleigh number 10^4 with 256^2 internal nodes, for the thermally driven cavity with insulated top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

(a)



(b)



(c)

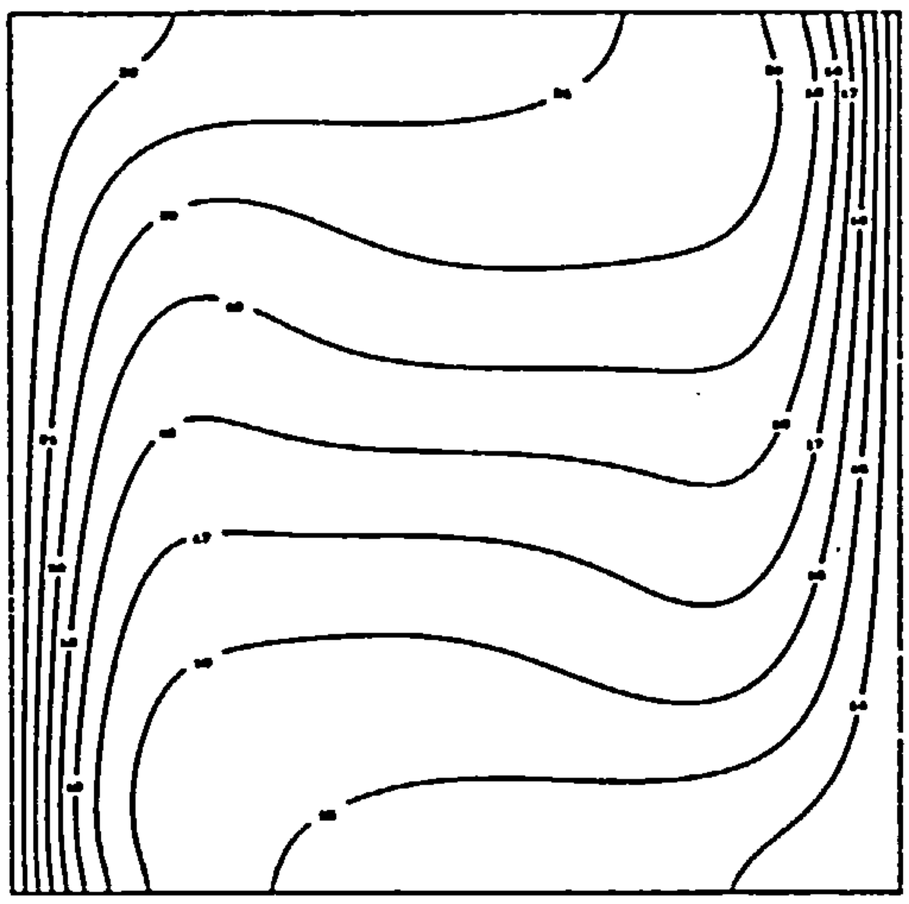
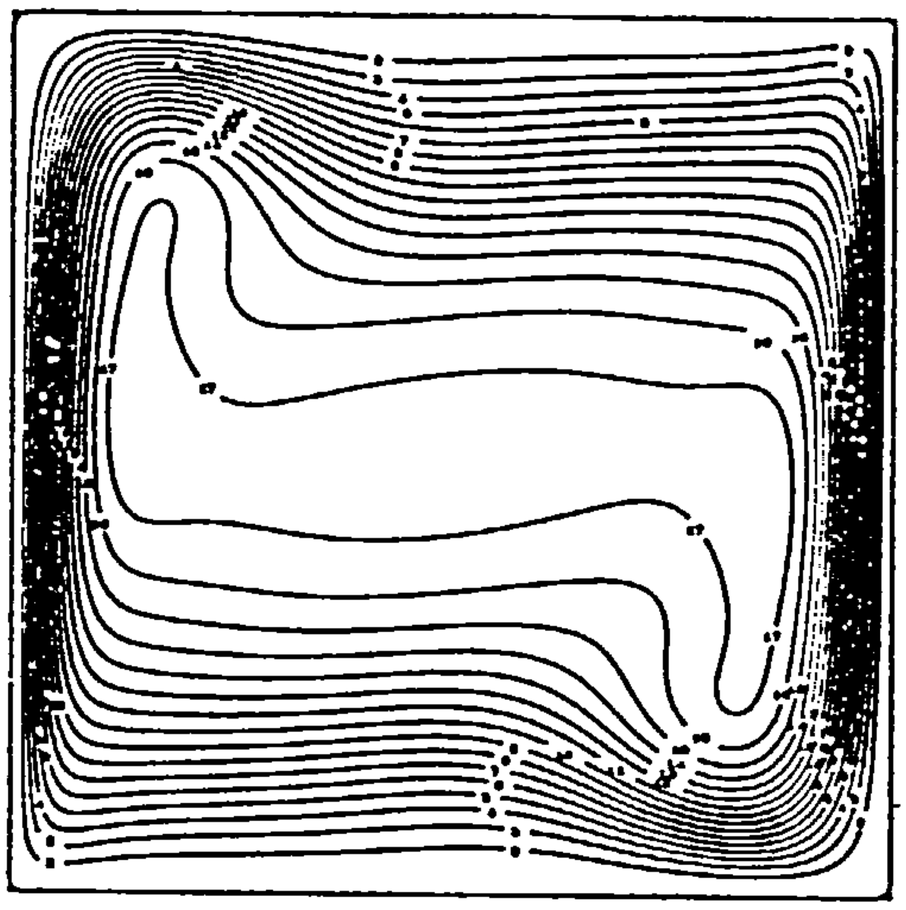


Figure 6.12: Contour plots for Rayleigh number 10^5 with 256^2 internal nodes, for the thermally driven cavity with insulated top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

(a)



(b)

(c)

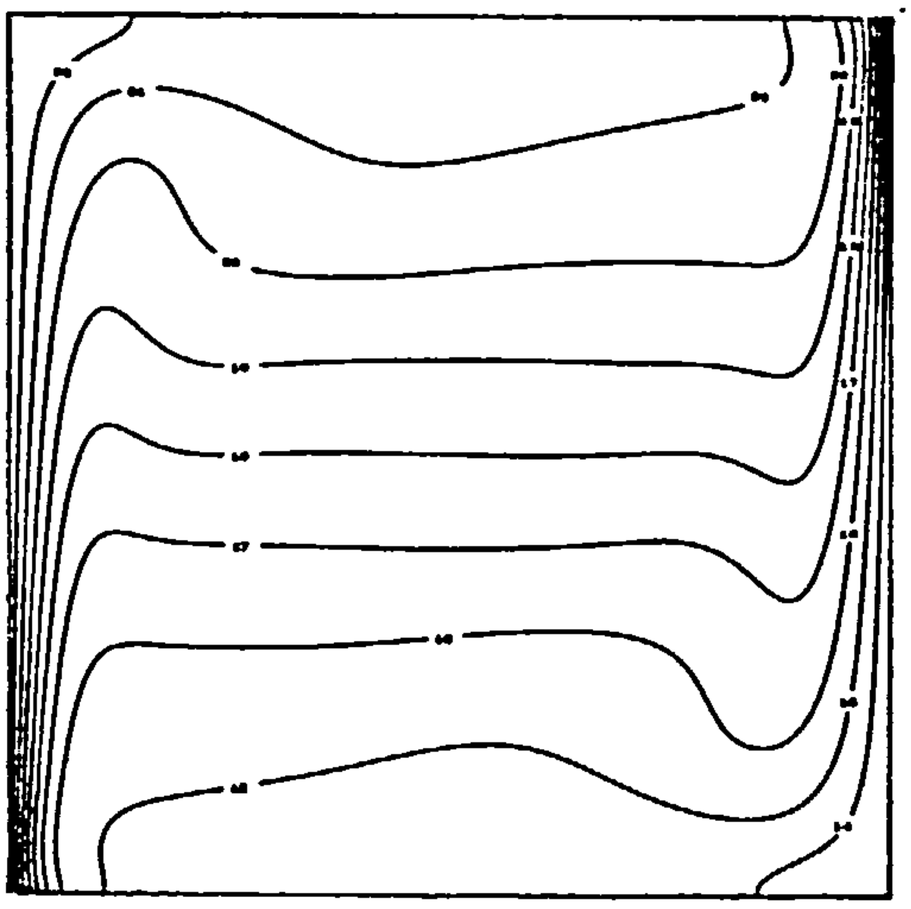


Figure 6.13: Contour plots for Rayleigh number 10^6 with 256^2 internal nodes, for the thermally driven cavity with insulated top and bottom walls: (a) Streamfunction, (b) vorticity and (c) temperature.

number of 0.71 on grids of upto 256^2 internal nodes. Boonkkamp observed that for Rayleigh number greater than 10^6 the solution was periodic. As before an increase in the Rayleigh number leads to separation of the streamfunction maximum into two, and the development of a region of slow flow and horizontal temperature contours. The zero derivative on the top and bottom boundaries results in a greater bunching of the temperature contours near the left and right boundaries. Gradients of temperature are even larger. The value of the maximum streamfunction is lower.

In 1983 de Vahl Davis[2] presented a benchmark solution for this problem. He used a streamfunction-vorticity formulation with central differencing and Richardson

Ra	10^3	10^4	10^5	10^6
ψ_{mid}	+0.03	-0.06	-0.3	+0.8
ψ_{max}	-	-	-0.2	+0.8
x	-	-	-1.4	+0.6
y	-	-	-0.5	0.0
U_{max}	+0.3	+0.3	-1.6	-0.7
y	+0.2	+0.1	-0.1	0.0
V_{max}	0.0	+0.02	-0.2	+0.01
x	0.0	+0.6	-1.5	+2.1
$\overline{Nu}$	-0.09	+0.4	+1.0	+0.8
Nu_{max}	-0.5	+1.9	+1.0	+0.8
y	+35.9	+9.0	+11.1	+3.2
Nu_{min}	-6.10	-2.4	-3.01	-0.9
y	0.0	0.0	0.0	0.0

Table 6.5: Percentage error of the present solution compared with that of de Vahl Davis.

	Rayleigh number			
Mesh	10^3	10^4	10^5	10^6
4^2	0.17	0.21	-	-
8^2	1.46	5.16	3.48	-
16^2	5.04	9.15	13.93	36.58
32^2	15.79	27.37	54.02	90.07
64^2	33.97	81.31	211.44	312.65
128^2	105.85	153.60	794.16	1114.52
256^2	391.68	439.71	3833.78	3902.44

Table 6.6: Computer time for the solution of the thermally driven cavity with adiabatic walls.

extrapolation. Several characteristics of the flow were given. Due to the difference in formulation and consequent differences in the non-dimensionalisation the results from equation (6.5) must be scaled as follows

$$u = u \times Gr^{½}Pr$$

$$v = v \times Gr^{½}Pr$$

In de Vahl Davis' work streamfunction was known and from this velocities were calculated. From temperature, heat flux was calculated as

$$Q(x,y) = u\theta - \frac{\partial\theta}{\partial x}. \qquad (6.10)$$

Before comparing the solutions found here with those of de Vahl Davis we should consider how they were obtained. When a maximum value was sought by de Vahl Davis, it and its location were calculated by numerical differentiation using a fourth order polynomial approximation. de Vahl Davis observed, "The interpolated values differed from the closest of the adjacent mesh point values by no more than 1 per cent in every case except one." In the present work only mesh point values were used. The grid in de Vahl Davis' work was non-staggered and based on a finite difference rather than a finite volume. Due to these factors is was not possible to calculate the heat flux with exactly the same operators as de Vahl Davis.

Table 6.5 shows percentage errors between the data in Table 6.4 and that of the benchmark solution. These show very close agreement between the two, particularly in view of the above comments, and the estimated error of the benchmark solutions given by de Vahl Davis as 0.1, 0.2, 0.3 and 1.0 for Rayleigh numbers of 10^3, 10^4, 10^5 and 10^6, respectively. The largest discrepancy is in the values of Nu_{max} and Nu_{min}. This is caused by particular difficulty in calculating $\frac{\partial T}{\partial x}$ at the boundaries. These errors compare well with the comparison exercise by de Vahl Davis[3]. They are of the same order as the other techniques. Over 30 methods were evaluated, using finite volumes, differences and elements, for both vorticity-streamfunction and primitive variable formulations,

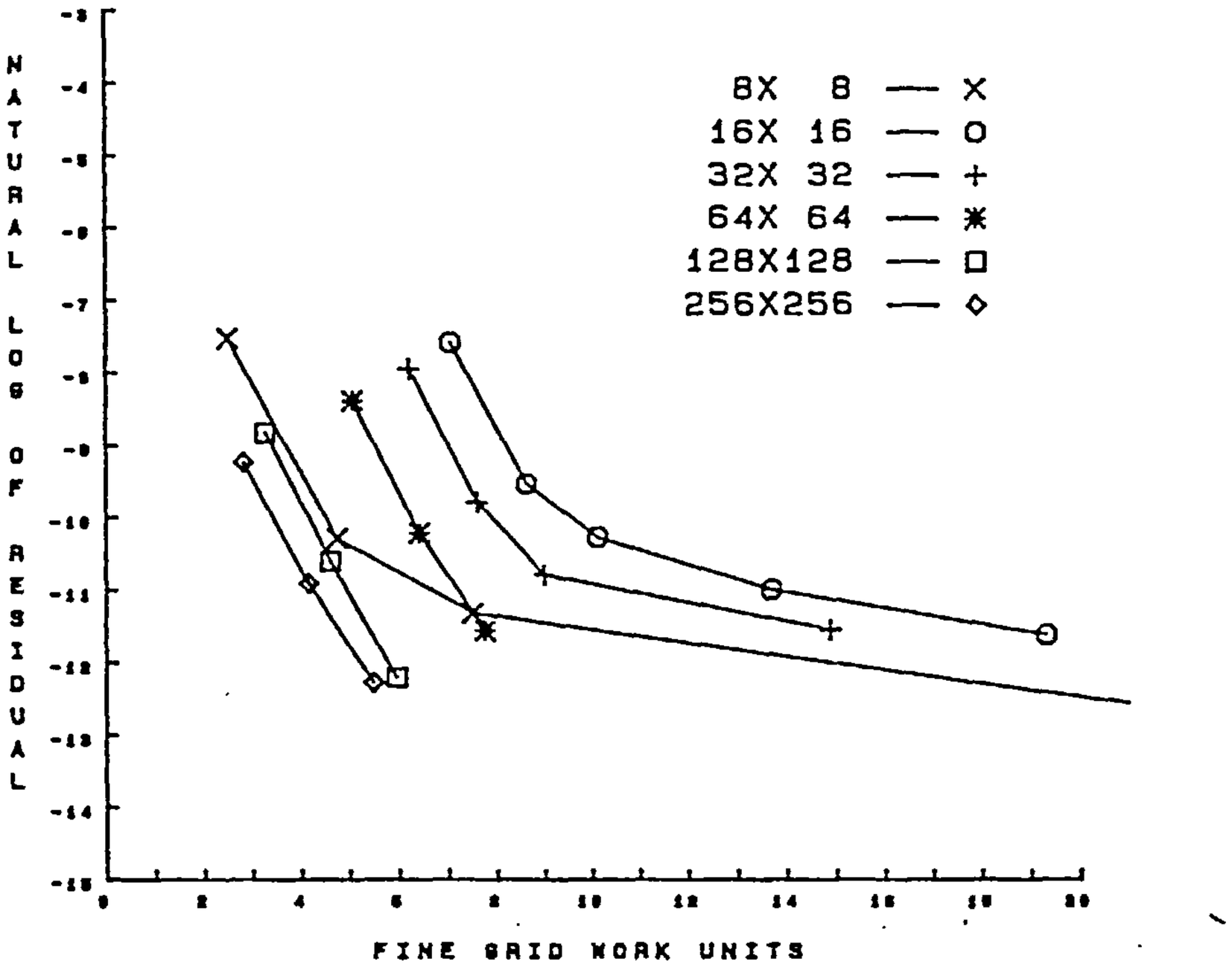


Figure 6.14: Plot of the natural log of the residual against FGWU for Rayleigh number 10^3 in the case of the thermally driven cavity with insulated top and bottom walls.

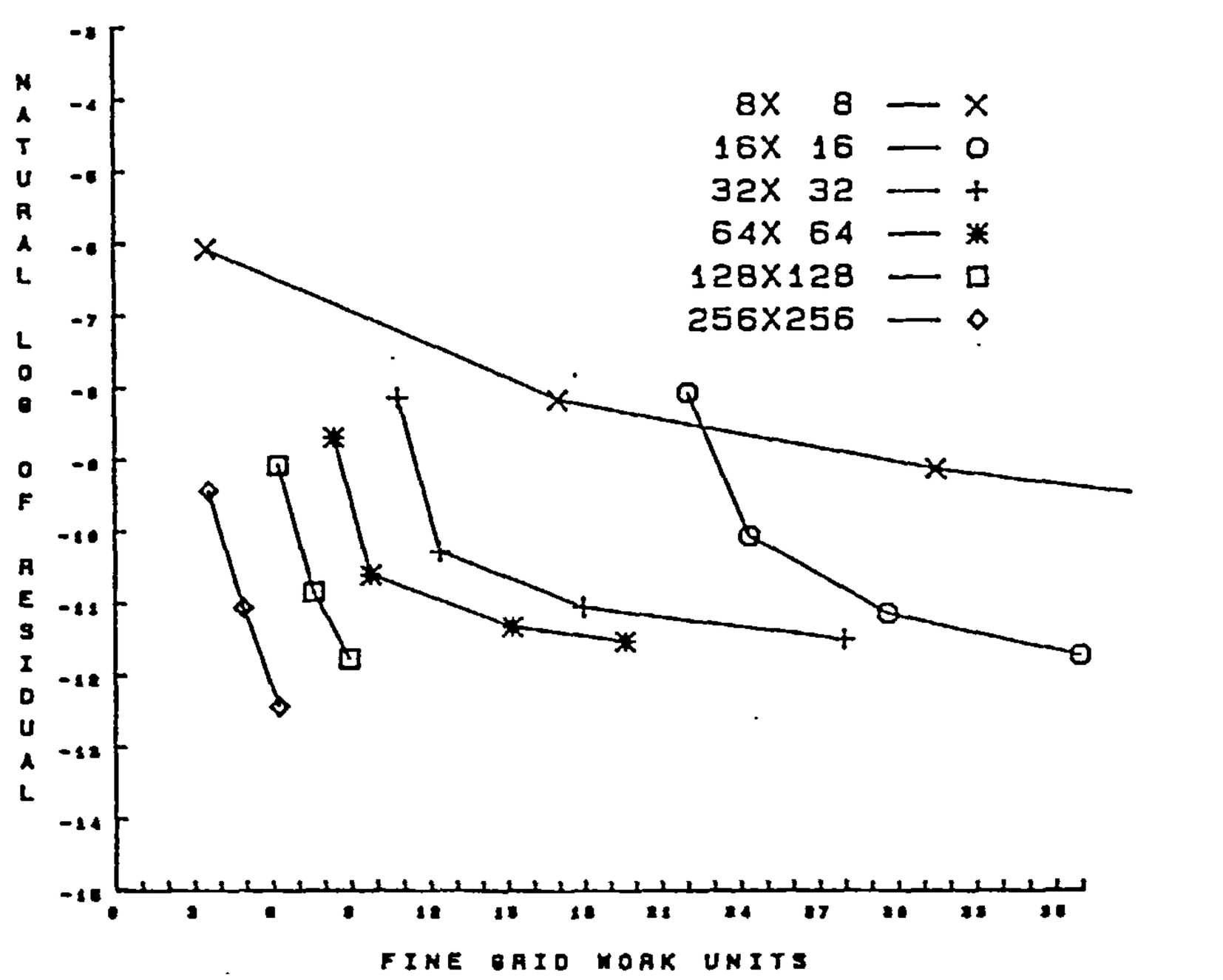


Figure 6.15: Plot.of the natural log of the residual against FGWU for Rayleigh number 10^4 in the case of the thermally driven cavity with insulated top and bottom walls.

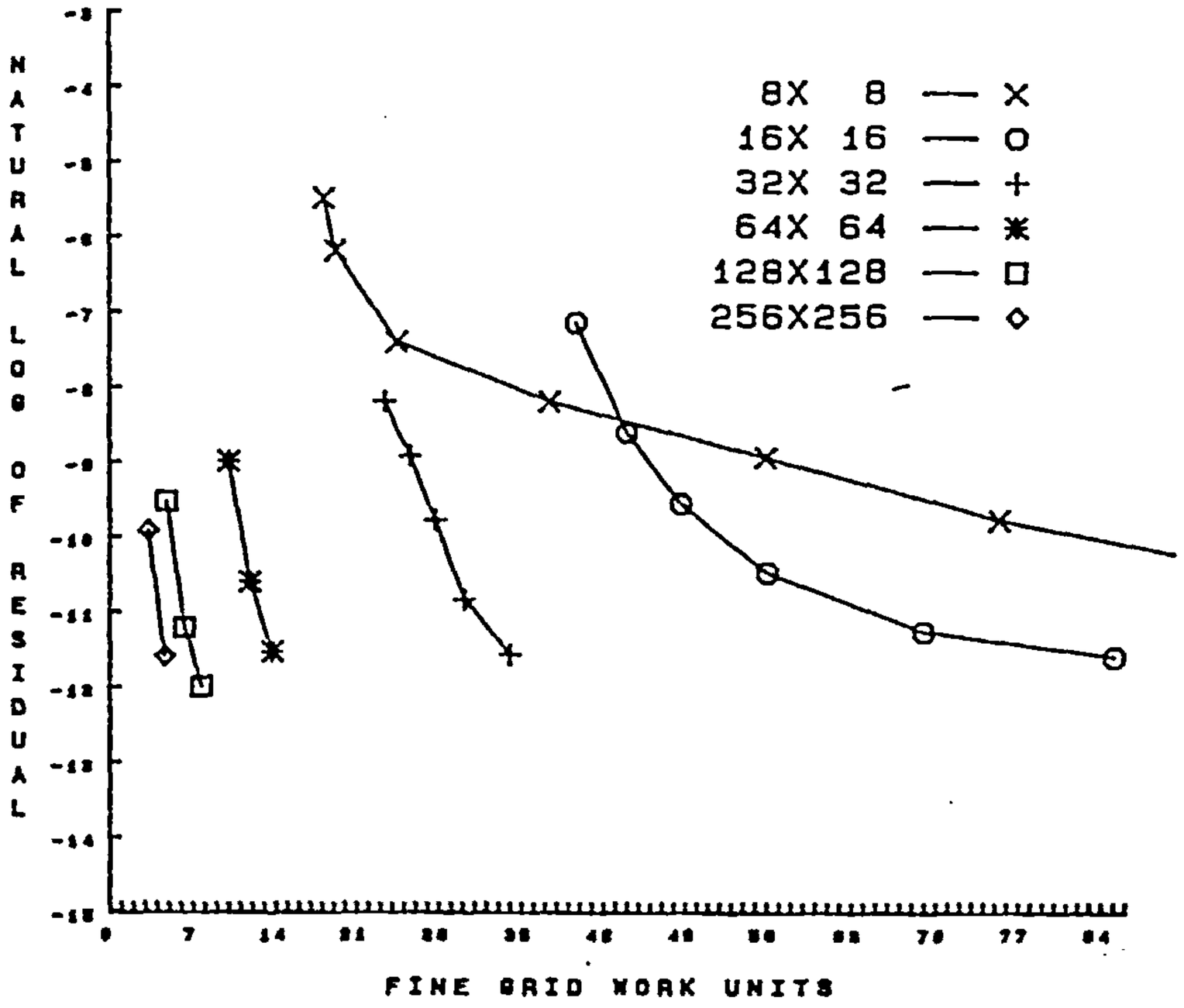


Figure 6.16: Plot of the natural log of the residual against FGWU for Rayleigh number 10^5 in the case of the thermally driven cavity with insulated top and bottom walls.

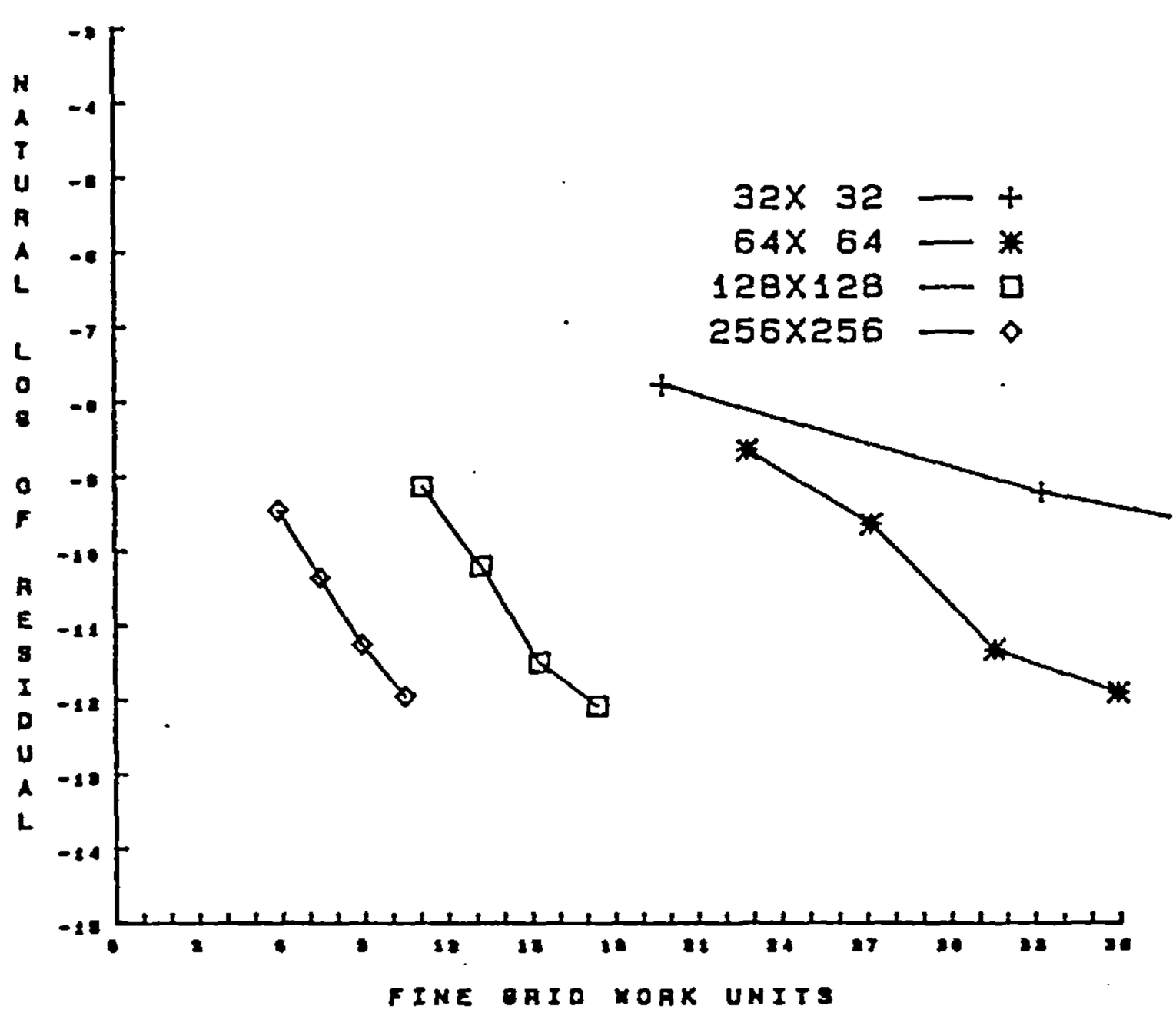


Figure 6.17: Plot of the natural log of the residual against FGWU for Rayleigh number 10^6 in the case of the thermally driven cavity with insulated top and bottom walls.

different orders of discretisation and various solution strategies. The largest number of finite difference nodes used was 65^2.

It can be seen that the method used here is efficient in producing accurate solutions in short times for thermal problems. For a solution on a 40^2 mesh at a Rayleigh number of 10^6 Boonkkamp's method used 3600 seconds on a Cyber computer. Here, solutions are achieved in a fraction of the time. These times are also much less then those of an earlier study[6]. This is due to an improved multigrid strategy and the use of multiple sweeps in the smoother. From Table 6.6 it can be seen that FGWU decreases as the number of internal nodes increases, so h-independence is achieved. From Figures 14-17it can be seen that multigrid convergence is achieved. Table 6.7 shows multigrid convergence rate as defined in equation(4.16). Overall these rates are slightly higher than for the cavity with perfectly conducting walls. This is probably a reflection of the use of derivative conditions, but as the number of grids increases the effect diminishes. So it seems that once optimal behaviour is approached the effect of the derivative conditions is minimal.

Ra	10^3	10^4	10^5	10^6
4^2	0.40	0.48	0.65	-
8^2	0.69	0.93	0.95	-
16^2	0.65	0.68	0.87	-
32^2	0.58	0.76	0.67	0.90
64^2	0.28	0.69	0.40	0.75
128^2	0.26	0.33	0.37	0.59
256^2	0.28	0.30	0.28	0.54

Table 6.7: Multigrid convergence rates for the cases considered in section 6.3

References

1. Boonkkamp, J.H.M. ten Thije, *The Odd-Even Hopscotch Pressure Correction Scheme for the Computation of Free Convection in a Square Cavity*, Centre for Mathematics and Computer Science (1980).

2. Davis, G. de Vahl, "Natural Convection of Air in a Square Cavity: A Bench Mark Numerical Solution", *International Journal for Numerical Methods in Fluids* 3 pp. 249-264 (1983).

3. Davis, G. de Vahl and Jones, I.P., "Natural Convection in a Square Cavity: A Comparison Exercise", *International Journal for Numerical Methods in Fluids* 3 pp. 227-248 (1983).

4. Galpin, P.F. and Raithby, G.D., "Numerical Solution of Problems in Incompressible Fluid Flow : Treatment of the Temperature-Velocity Coupling", *Numerical Heat Transfer* 10 pp. 105-129 (1986).

5. Gaskell, P.H. and Lau, A.K.C., "Curvature Compensated Convective Transport : SMART, a New Boundedness Preserving Transport Algorithm", *International Journal for Numerical Methods in Fluids* 8 pp. 617-641 (1988).

6. Gaskell, P.H. and Wright, N.G., "Multigrid Algorithm for the Investigation of Thermal, Recirculating Fluid Flow Problems", in *Numerical Methods in Thermal Problems, Proceedings of The Fifth International Conference,* , Montreal, Canada (July 1987).

7. Launder, B.E. and Spalding, D.B., "The Numerical Computation of Turbulent Flows", *Computer Methods in Applied Mechanics and Engineering* 3 pp. 269-289 (1974).

8. Luchini, P., "An Adaptive-Mesh Finite-Difference Solution Method for the Navier-Stokes Equations", *Journal of Computational Physics* 68 pp. 283-306 (1987).

9. Priddin, C., "Private Communication", *Rolls-Royce,p.l.c., Derby*, (1987).

Chapter 7

CONCLUSION

A multigrid scheme has been presented that uses an unsegregated smoother with a higher order approximation to convective transport. This novel approach has been applied to a variety of test cases. On the whole the results from these are encouraging and give reason to believe that multigrid techniques are viable and useful in Computational Fluid Dynamics.

The solution times for the various test cases are much lower than could be obtained with traditional methods. This shows that multigrids enable us to find solutions to complex problems, and problems in three-dimensions that would not have been possible before. The emphasis has been to find solutions to problems of engineering interest, while bearing in mind theoretical considerations. Even though an in depth theoretical investigation was not carried out, h-independent convergence was achieved and the convergence rates compare well with theoretical ones, contained in the literature, for multigrid techniques generally. For the cases considered here the numbers of nodes used are equal to, or larger, than those used in other work, and the solutions obtained must be considered to be amongst some of the most accurate available.

As is often the case, the present work throws up as many questions and problems as it answers! One problem is the complexity of the method, which is an important consideration with respect to its incorporation in industrial codes. However, this must be weighed against the reduction in cost from its use. One advantage of the sensitivity of multigrid to incorrect implementation, is that it can often detect (by exhibiting degraded convergence) unphysical aspects of the discretisation or problem formulation.

The BIM used here is not ideally suited for the large computational molecule of higher order schemes such as CCCT. This means that low relaxation factors must be used as the non-linear effects increase. As CCCT represents by far the best option for discretisation of the equations, this problem should be investigated further. One remedy for this would be to use a superior treatment of the non-linear terms when discretising the Navier-Stokes equations[3]. This should improve convergence by taking

greater account of coupling between variables. The principle answer would be to develop a solution technique well suited to CCCT. This could well be a line-by-line solver as opposed to a point-by-point one. Such solvers have been observed to be well suited to higher-order differencing, because they treat the equations in a fully implicit manner along the line[6]. They are particularly good at reducing errors with a wavelength similar to the meshsize[6], which is just the property we require of a smoother. Another avenue worth pursuing, but rather more radical than the above in terms of conventional work in this area, is that of using an explicit time-dependent or false transient solution technique. These have no problem in deciding the order of solution over the domain as all calculations are based on old values. The residuals required with multigrid are readily available for use, requiring no extra calculation as with semi-implicit schemes (BIM, CELS, SIMPLE). Such a technique could be simply and efficiently adapted for solution in a vector processor or transputer, both of which are becoming increasingly available. The problem of trying to find steady state solutions at high values of Reynolds number or Rayleigh number to problems that may well be periodic (as can happen with the backward facing step and the thermal cases investigated here) would not occur. In fact, such solutions should be easily obtained and examined.

The staggered grid used here and in many other works, generates its own problems. One is the specification of outflow boundary conditions and the effect of this on multigrids, as found in Chapter 5. The boundary treatment in general is extremely cumbersome and requires special consideration for multigrids. It is also very problematic from the aspect of constructing and programming restriction and prolongation operators. The multigrid strategy may well work better if a non-staggered formulation were adopted. Several authors[5, 2, 1] have used such a grid with primitive variables and obtained good results.

An alternative grid arrangement could incorporate the idea of Lonsdale, whose grid coarsening maintains the near-boundary point at a constant distance from the boundary while varying meshsizes in the interior. This improves the consistency of

boundary treatment between grids.

The problems encountered with singularities in Chapter 5, could be alleviated by the use of an analytical solution in the region near to a singularity. This idea has been used elsewhere[4], but may need further attention for use with multigrids.

Further points worth investigation are the use of a Newton solver for the coarsest grid problem, and the implementation of CCCT(variable α) with multigrid giving particular attention to the transfer of the α's between grid levels.

Investigation is also required into the relationship between grids with regard to interpolation and boundary conditions in multigrids and also into the possibility of a closer correspondence between interpolation in the restriction and prolongation operators and interpolation in the control volume discretisation.

References

1. Burns, A.D., Wilkes, N.S., Jones, I.P., and Kightley, J.R., *FLOW3D: Body-Fitted Coordinates, AERE-R 12262*, Computer Science and Systems Division, Harewell Laboratory (1986).

2. Chorin, A.J., ''Numerical Solution of the Navier-Stokes Equations'', *Mathematics of Computation* **22** p. 745 (1968).

3. Galpin, P.F. and Raithby, G.D., ''Treatment of Non-linearities in the Numerical Solution of the Incompressible Navier-Stokes Equations'', *International Journal for Numerical Methods in Fluids* **6** pp. 409-426 (1986).

4. Kelmanson, M.A., ''Boundary Integral Equation Analysis of Singular Potential and Biharmonic Problems'', in *Ph.D. Thesis*, University of Leeds (1983).

5. Linden, J., Steckel, B., and Stuben, K., ''Parallel Multigrid Solution of the Navier-Stokes Equation on General 2D-Domains'', *Arbeitspapiere der GMD*, (294)(1988).

6. Miller, T.F. and Schmidt, F.W., ''Evaluation of a Multilevel Technique Applied to the Poisson and Navier-Stokes Equations'', *Numerical Heat Transfer* **13** pp.

1-26 (1988).

APPENDICES

Appendix I - Interpolation Operators

The two-dimensional interpolation operators are as follows.

Restriction of a Scalar

This operator is used for the restriction of pressure and the scalars stored at the centre of the control volumes. Here, it is shown for pressure. It sweeps through the domain considering each i and j value in turn. If i_c and i_c are the i and j values of the point on the coarse grid, then i_f and j_f, the i and j values on the fine grid, are defined as

$$i_f = 2i_c, \quad j_f = 2j_c.$$

The equation for $p(i_c,j_c)$ is

$$p(i_c,j_c) = \frac{p(i_f,j_f)+p(i_f,j_f-1)+p(i_f-1,j_f)+p(i_f-1,j_f-1)}{4}.$$

Restriction of Velocity

This is used for the two velocity components. With i_c, j_c, i_f and j_f defined as above

$$u(i_c,j_c) = \frac{u(i_f,j_f)+u(i_f,j_f-1)}{2}.$$

$$v(i_c,j_c) = \frac{v(i_f,j_f)+v(i_f-1,j_f)}{2}.$$

Prolongation of a Scalar

Sweeping through the domain as above, with $i_f = 2i_c-1$ and $j_f = 2j_c-1$ then

$$p(i_f,j_f) = \frac{9p(i_c,j_c)+3(p(i_c-1,j_c)+p(i_c,j_c-1))+p(i_c-1,j_c-1)}{16}$$

$$p(i_f,j_f-1) = \frac{9p(i_c,j_c-1)+3(p(i_c-1,j_c-1)+p(i_c,j_c))+p(i_c-1,j_c)}{16}$$

$$p(i_f-1,j_f) = \frac{9p(i_c-1,j_c)+3(p(i_c-1,j_c-1)+p(i_c,j_c))+p(i_c,j_c-1)}{16}$$

$$p(i_f-1,j_f-1) = \frac{9p(i_c-1,j_c-1)+3(p(i_c,j_c-1)+p(i_c-1,j_c))+p(i_c,j_c)}{16}.$$

Prolongation of Velocity

With i_f, j_f, i_c and j_c defined as for pressure

$$u(i_f,j_f) = \frac{3u(i_c,j_c)+u(i_c,j_c-1)}{4}$$

$$u(i_f,j_f-1) = \frac{3u(i_c,j_c-1)+u(i_c,j_c)}{4}$$

$$u(i_f-1,j_f) = \frac{3(u(i_c,j_c)+u(i_c-1,j_c))+u(i_c,j_c-1)+u(i_c-1,j_c-1)}{8}$$

$$u(i_f-1,j_f-1) = \frac{3(u(i_c,j_c-1)+u(i_c-1,j_c-1))+u(i_c,j_c)+u(i_c-1,j_c)}{8},$$

and

$$v(i_f,j_f) = \frac{3v(i_c,j_c)+v(i_c-1,j_c)}{4}$$

$$v(i_f-1,j_f) = \frac{3v(i_c-1,j_c)+v(i_c,j_c)}{4}$$

$$v(i_f,j_f-1) = \frac{3(v(i_c,j_c)+v(i_c,j_c-1))+v(i_c-1,j_c)+v(i_c-1,j_c-1)}{8}$$

$$v(i_f-1,j_f-1) = \frac{3(v(i_c-1j_c)+v(i_c-1,j_c-1))+v(i_c,j_c)+v(i_c,j_c-1)}{8}.$$

The interpolation for three-dimensions is the obvious extension of the two-dimensional case.

Appendix II -Solution of a banded matrix

Consider the matrix system

$$\begin{bmatrix} a_1 & 0 & 0 & 0 & c_1 \\ 0 & a_2 & 0 & 0 & c_2 \\ 0 & 0 & a_3 & 0 & c_3 \\ 0 & 0 & 0 & a_4 & c_4 \\ b_1 & b_2 & b_3 & b_4 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix}.$$

The matrix can be decomposed into a lower and upper form, LU

$$\begin{bmatrix} a_1 & 0 & 0 & 0 & c_1 \\ 0 & a_2 & 0 & 0 & c_2 \\ 0 & 0 & a_3 & 0 & c_3 \\ 0 & 0 & 0 & a_4 & c_4 \\ b_1 & b_2 & b_3 & b_4 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} m_1 & 0 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 & 0 \\ 0 & 0 & m_3 & 0 & 0 \\ 0 & 0 & 0 & m_4 & 0 \\ l_1 & l_2 & l_3 & l_4 & m_5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & u_1 \\ 0 & 1 & 0 & 0 & u_2 \\ 0 & 0 & 1 & 0 & u_3 \\ 0 & 0 & 0 & 1 & u_4 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} m_1 & 0 & 0 & 0 & m_1u_1 \\ 0 & m_2 & 0 & 0 & m_2u_2 \\ 0 & 0 & m_3 & 0 & m_3u_3 \\ 0 & 0 & 0 & m_4 & m_4u_4 \\ l_1 & l_2 & l_3 & l_4 & l_1u_1+l_2u_2+l_3u_3+l_4u_4+m_5 \end{bmatrix}.$$

So for i = 1 to 4

$$l_i = b_i$$

$$m_i = a_i$$

$$u_i = \frac{c_i}{m_i}$$

and

$$m_5 = -\sum_{i=1}^{4} l_i u_i.$$

Having solved these equations the x_i are calculated as follows

Obtain the z_i by

for i = 1 to 4

$$z_i = \frac{ri}{m_i}$$

$$z_5 = -\sum_{i=1}^{4} \frac{ric_i}{m_5}.$$

Then obtain the x_i by

$$x_5 = z_5$$

for i = 4 to 1

$$x_i = z_i - u_i z_5.$$

The analogous technique for a 7x7 matrix is derived as above, trivially.