SIMULATION OF TURBULENT FLOWS USING NODAL INTEGRAL METHOD

BY

SUNEET SINGH

B.Eng., Bhopal University, 1998
M.Tech., Indian Institute of Technology Kanpur, 2000

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Nuclear Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Doctoral Committee:

Prof. Rizwan-uddin, Chair
Prof. Barclay G. Jones
Prof. Roy A. Axford
Prof. Surya Pratap Vanka

SIMULATION OF TURBULENT FLOWS USING NODAL INTEGRAL METHOD

Suneet Singh, Ph.D.
Department of Nuclear, Plasma and Radiological Engineering
University of Illinois at Urbana-Champaign, 2008
Rizwan-uddin, Adviser

Nodal methods are the backbone of the production codes for neutron-diffusion and transport equations. Despite their high accuracy, use of these methods for simulation of fluid flow is relatively new. Recently, a modified nodal integral method (MNIM) has been developed for simulation of laminar flows. In view of its high accuracy and efficiency, extension of this method for the simulation of turbulent flows is a logical step forward. In this dissertation, MNIM is extended in two ways to simulate incompressible turbulent flows—a new MNIM is developed for the 2D k-ε equations; and 3D, parallel MNIM is developed for direct numerical simulations. Both developments are validated, and test problems are solved.

In this dissertation, a new nodal numerical scheme is developed to solve the k-ε equations to simulate turbulent flows. The MNIM developed earlier for laminar flow equations is modified to incorporate eddy viscosity approximation and coupled with the above mentioned schemes for the k and ε equations, to complete the implementation of the numerical scheme for the k-ε model. The scheme developed is validated by comparing the results obtained by the developed method with the results available in the literature obtained using direct numerical simulations (DNS). The results of current simulations match reasonably well with the DNS results. The discrepancies in the results are mainly due to the limitations of the k-ε model rather than the deficiency in the developed MNIM.

A parallel version of the MNIM is needed to enhance its capability, in order to carry out DNS of the turbulent flows. The parallelization of the scheme, however, presents some unique challenges as dependencies of the discrete variables are different from those that exist in other schemes (for example in finite volume based schemes). Hence, a parallel MNIM (PMNIM) is developed and implemented into a computer code with communication strategies based on the above mentioned dependencies. The speedup and efficiency of the PMNIM are analyzed for a laminar flow test problem. .The efficiency, calculated based on Gustafson's law, is found to be more than 75% for a $20 \times 20 \times 20$ mesh and remains almost constant as number of processors is increased. It can be concluded that the PMNIM is reliable, scalable and efficient.

The PMNIM is then used to study the transition to turbulence in Arnold-Beltrami-Childress (ABC) flows. These flows display the interesting phenomenon of heteroclinic cycles. The results are obtained for two wavenumbers: $k = 1$ (also studied earlier by other researchers) and $k = 2$, respectively. The results for $k = 1$ are compared with those obtained using the pseudo spectral method. The comparison shows good agreement and also shows that results obtained with similar grid sizes and time steps match very well with those obtained using pseudo spectral method. New results are obtained for $k = 2$, and the heteroclinic cycles observed in this flow are discussed and contrasted with those obtained in the flow with $k = 1$. The results show that the flow becomes unstable for the $k = 2$ case at smaller Reynolds number than that for $k = 1$. The flow also shows some very interesting phenomena such as simultaneous existence of the two types of heteroclinic cycles.

Acknowledgements

I wish to take this opportunity to express my sincere gratitude towards my advisor Professor Rizwan-uddin for his valuable support throughout the project work. Without his insight, guidance and encouragement; this work would not have taken its present shape. I would also like to thank Professor Roy A. Axford, Professor Barclay G. Jones and Prof. Surya Pratap Vanka for serving in the doctoral committee for this dissertation.

I would like to thank Allen Toreja and Fei Wang for providing help with the modified nodal integral method. My thanks are due to Prashant Jain, Federico Teruel, Surya Kumar Saripella, Rajneesh Chaudhary and Hitesh Bindra for several fruitful discussions during the course of this dissertation. Finally, I would like to acknowledge all the faculty and staff of NPRE for their support and guidance.

Suneet Singh

Table of Contents

List of Figures

List of Tables

List of Acronyms

ABC	Arnold-Beltrami-Childress
AKN	Abe Kondoh Nagano
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
FPSM	Fourier Pseudo Spectral Method
JFNK	Jacobian Free Newton Kyrlov
MNIM	Modified Nodal Integral Method
MPI	Message Passing Interface
NIM	Nodal Integral Method
PetSc	Portable extensible Toolkit for Scientific Computation
PMNIM	Parallel Modified Nodal Integral Method
RANS	Reynolds Averaged Navier-Stokes
SOR	Successive Over-relaxation
TIP	Transverse Integration Procedure

Chapter 1

Introduction

1.1 Motivation

Despite recent progress in Computational Fluid Dynamics (CFD) (Tannehill 1997, Morton 1994), simulation of many flows of practical interest is still beyond present capabilities. This is especially true for turbulent flows. Since many flows of practical interest are turbulent, it is necessary to strive for better computational schemes and better turbulence modeling to tackle the problem.

A number of *coarse-mesh* schemes have been developed to solve problems in large domains. These schemes yield numerical solution with comparable accuracy in less CPU time. Nodal methods, which are a subset of *coarse-mesh* methods, have been successfully developed to numerically solve the laminar flow problems. The substantial savings in CPU time serve as an encouragement for developing Nodal Integral Methods (NIM) for modeling of turbulent flows.

Several turbulence models have been used with different numerical schemes with varying degree of success. The turbulence model most commonly employed for practical applications is the k-ε model. With the advent of High Performance Computing there has been considerable progress in turbulence modeling using more computer-intensive models. The Large-Eddy Simulations (LES) of turbulence have, for example, become

popular with the recent increases in computing power. Direct Numerical Simulation (DNS) of flows has also progressed to higher Reynolds numbers and relatively complex geometries.

Combining recent advances in nodal numerical schemes and parallel computing promises an efficient path toward significantly enhancing our capabilities to simulate turbulent flows. The aforementioned advances in the nodal integral methods are exploited in this dissertation to enhance the capability of turbulent flow simulations using the k- ε turbulence model as well as using DNS of turbulent flow.

A nodal integral method is developed for the high Reynolds number k-ε model as well as for the low Reynolds number k-ε model and implemented in a computer code. The developed methods are used for simulation of some simple flows.

The parallel version of the MNIM for three-dimensional incompressible Navier-Stokes and energy equations is also developed so as to make the method capable of direct numerical simulation of turbulent flows. Simulations are carried out for some interesting flows using the developed method.

A brief background of nodal integral methods is presented in the following section. A brief review of k-ε model and DNS are given in the subsequent sections in the context of the work carried out in this dissertation.

1.2 Nodal Integral Method

Nodal methods, first developed for multi-group neutron diffusion and neutron transport equations, now constitute the backbone of production codes used in the nuclear industry. An early review of nodal methods, developed and used by the nuclear industry,

is given by Lawrence (1986). Nodal schemes are developed by approximately satisfying the governing differential equations on finite size brick-like elements that are obtained by discretizing the space of independent variables.

Similar approaches have been used in other branches of science and engineering to develop efficient numerical schemes (Wescott 2001, Horak 1985, Elnawawy 1990). Nodal methods, as a general class of computational schemes, are discussed by Hennart (1986). Nodal integral methods (NIM), a subclass of nodal methods, have been developed for the steady-state (Azmy 1983) and time-dependent (Wilson 1988) Navier–Stokes equations. NIM was applied to the steady-state Boussinesq equations for natural convection, and to several steady-state incompressible flow problems (Azmy 1985). Esser (1993) developed a nodal scheme for the two-dimensional, vorticity-stream function formulation of the Navier–Stokes equations. This development – that leads to inherent upwinding in the numerical scheme – however cannot be easily extended to three dimensions. NIM was also developed and applied to the time-dependent heat conduction problem (Wilson 1988). Michael (2001) developed a second- and a third-order NIM for the convection–diffusion equation.

Though highly innovative, those early applications of nodal methods for the Navier-Stokes equations did not take full advantage of the potential that the nodal approach offers. An improved method has recently been developed to solve the incompressible Navier-Stokes equations. This modified nodal integral method (MNIM) was first developed for 2D, time dependent problems (Wang 2003), and then extended for 3D, time dependent flows as well (Wang 2005). However, since NIMs rely on the transverse-integration procedure, they are therefore restricted to physical domains with

boundaries parallel to one of the axis, i.e, to geometries that can be filled with brick-like cells. Recently, methods have been developed, for fluid flow (Toreja 2003a, Toreja 2003b) as well as for neutron diffusion equations (Gu 2005), combining NIM with other numerical methods (for example finite element method and finite analytic method) in order to overcome this restriction. Recently developed MNIM is used for simulation of the turbulent flows in the present work. A brief review of the MNIM is given in Chapter 2. Some modifications in the formulation of the boundary conditions are also discussed in the abovementioned chapter.

The recent developments in NIMs increase its capability to handle more complex flows. The extension of NIM to simulate turbulent flows is therefore a logical step forward. In the present work, MNIM is developed for the k-ε model and used to simulate turbulent flows. In addition, MNIM is also extended to carry out direct numerical simulations (DNS) and has been used for the same.

1.3 k-ε Model

The k-ε model, a two equation model, is the most commonly used model for the simulation of the turbulent flows (Durbin 2001). The widespread use of this model can be attributed to the fact that it produces *reasonably* good results with relatively less computational effort. Several variations of these models are developed and implemented in the last four decades. Although this model has several limitations, relatively much more extensive computational requirements for alternative models and approaches, make the k-ε based approaches fairly popular. It is expected that it will remain the method of

choice in the near future. This is especially true for complex engineering flows. The method, including its various variations has been reviewed in Chapter 3.

The k- ε model implementation is carried out by incorporating eddy viscosity in the recently developed Modified Nodal Integral Method (MNIM) for incompressible Navier-Stokes equations. The NIM solver for k and ε equations are also developed and incorporated into a computer code. The flow simulations for some simple flows are carried out to validate the computer code. The implementation of the model and results are discussed in Chapter 3.

.

1.4 Direct Numerical Simulation

Direct Numerical Simulation, though limited to low Reynolds numbers, is useful for study of physics of turbulence (Moin 1998). Moreover, DNS has resulted in improvement of k- ε and LES models. Therefore, one can expect DNS at higher Reynolds numbers and complex geometries to help in further improvement of turbulence models.

The presence of steep gradients and fluctuating velocities in turbulent flows makes it necessary to have large number of grid points in the computational domain. Therefore, with current computational capabilities DNS is feasible only with numerical schemes of higher order or special numerical methods (e.g. spectral methods) in which number of grid points is relatively less. Due to extensive computational cost of the DNS of turbulent flows, the MNIM needs to be parallelized to allow simulation of such flows.

1.4.1 Parallel MNIM

Several parallelization strategies have been developed for unstructured as well as structured grids for different numerical methods. These have been implemented on various types of machines. A brief survey of the parallelization schemes for fluid flow solvers based on various numerical schemes is given in Chapter 4.

The parallelization of MNIM has been carried out using Message Passing Interface (MPI) (Gropp 1997). The domain decomposition paradigm has been used for the parallel version of the MNIM (PMNIM). In order to develop a parallelization strategy for the method, various dependencies of the discrete variables on each other in the MNIM are identified,. Based on the dependencies of the variables, communication procedures are implemented using the MPI. The speedup and efficiency of the PMNIM are obtained for a test problem. The analysis of the parallelization scheme is done based on Gustafson's law (Gustafson 1988) as well as Amdahl's Law (Amdahl 1967). The details of the development and implementation of the PMNIM are discussed in Chapter 4.

The developed PMNIM has been used for the simulation of transition in the Arnold-Beltrami-Childress (ABC) flows.

1.4.2 Arnold-Beltrami-Childress (ABC) Flows

ABC flows have been recently studied, numerically as well as analytically, as a generator of turbulence (Podvigina 1994, Podvigina 2006). The large symmetry group present in the governing equations of the ABC flow results in the generation of the heteroclinic cycles at high Reynolds numbers. These cycles are obtained due to the symmetry breakdown caused by non-linear instabilities. A review of the heteroclinic

cycles and the ABC flows is given in Chapter 5. The chapter also presents a survey of the developments in the DNS of fully turbulent flows as well as DNS of the transition to the turbulence. The simulations are carried out for the transitions to turbulence in the ABC flows with wavenumber $k = 1$. The results are compared with the results obtained by Podvigina (1994) using pseudo-spectral method. New results are obtained for the ABC flow with wavenumber $k = 2$. Detailed results of these simulations are given in Chapter 5.

Chapter 6 summarizes the conclusions of this dissertation. The chapter also discusses the capabilities and limitations of the MNIM in the context of the simulation of the turbulent flows. Ideas and motivation for future work are also presented so as to enhance the capabilities of the method and to overcome some of its limitations.

Chapter 2

Review of Modified Nodal Integral Method for the Navier Stokes Equations

A brief description of the MNIM as applied to 3-D, time dependent, incompressible Navier-Stokes equations is presented here. Further details of the development can be found in Wang (2003) and Wang (2005). In addition to the review of MNIM, some modifications in the formulation of pressure boundary conditions are discussed. This chapter also discusses the formulation of Robin (or mixed) boundary conditions developed originally by Michael (2001) for convection diffusion equation. The equation is reformulated in the present work to be consistent with the MNIM.

As a prototypic equation for transfer processes involving diffusion and advection/convection, the review starts with the description of the nodal scheme for the convection-diffusion equation for the variable *C*.

2.1 NIM for the Convection Diffusion Equation

The space and time domain (*X, Y, Z, T*) is discretized in parallelepiped cells (*i, j, k, n*) of size $(2a_i \times 2b_j \times 2c_k \times 2\tau_n)$ with cell centered local coordinates

$(x, y, z, t; -a_i \le x \le a_i, -b_j \le y \le b_j, \quad -c_k \le z \le c_k, -\tau_n \le t \le \tau_n)$. The convection-diffusion equation in a cell is written as follows,

$$\frac{\partial C(x,y,z,t)}{\partial t} + u(x,y,z,t)\frac{\partial C(x,y,z,t)}{\partial x} + v(x,y,z,t)\frac{\partial C(x,y,z,t)}{\partial y} + w(x,y,z,t)\frac{\partial C(x,y,z,t)}{\partial z} = \bar{D}\left(\frac{\partial^2 C(x,y,z,t)}{\partial x^2} + \frac{\partial^2 C(x,y,z,t)}{\partial y^2} + \frac{\partial^2 C(x,y,z,t)}{\partial z^2} \right) + s(x,y,z,t) \tag{2.1}$$

where, u, v, and w are velocities in x, y and z directions, and $\bar{D}$ is the cell averaged diffusion coefficient. The above equation is valid for all the cells.

The next step in the NIM is the Transverse Integration Procedure (TIP). The TIP involves local averaging of the PDE over the cell in all independent variables except one, which results in a corresponding ODE. This process is repeated for all independent variables yielding 3 transverse integrated ODEs in the space variables, and one ODE in time. For example, averaging over y, z and t i.e. operating by $\frac{1}{8b_j c_k \tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k} dydzdt$, one gets,

$$\bar{D}\frac{d^2\bar{C}^{yzt}(x)}{dx^2} - \bar{u}\frac{d\bar{C}^{yzt}(x)}{dx} = \bar{S}^{yzt}(x) \tag{2.2}$$

where,

$$\bar{C}^{yzt}(x) \equiv \frac{1}{8b_j c_k \tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k} C(x,y,z,t)dydzdt \tag{2.3}$$

and the pseudo-source term $\bar{S}^{yzt}(x)$ is defined as

$$\overline{S}^{yzt}(x) \equiv \frac{1}{8b_j c_k \tau_n} \int_{-\tau_n}^{+\tau_n} \int_{-b_j}^{+b_j} \int_{-c_k}^{+c_k} \left(\begin{array}{l} \bar{D}\left(-\frac{\partial^2 C}{\partial y^2} - \frac{\partial^2 C}{\partial z^2} \right) - s(x,y,z,t) \\ +\bar{v}\frac{\partial C}{\partial y} + \bar{w}\frac{\partial C}{\partial z} + \frac{\partial C}{\partial t} \end{array} \right) dydzdt \qquad (2.4)$$

Here, $\bar{u}$, $\bar{v}$ and $\bar{w}$ are cell averaged velocities, and average of the product has been approximated by product of the averages, known to be a second order approximation. The above development is valid for all the cells.

The ODEs are solved after the pseudo-source terms are expanded in, say, Legendre polynomials and truncated at the zeroth order, which leads to a second order scheme. (Truncating at orders higher than zero leads to 3rd and higher order schemes.) The solution in the node (*i, j, k, n*) is given by

$$\overline{C}^{yzt}_{i,j,k,n}(x) = c_1 \exp\left(\frac{\bar{u}_{i,j,k,n} x}{\bar{D}_{i,j,k,n}} \right) + c_2 - \frac{\overline{S}^{yzt}_{0i,j,k,n}}{\bar{u}_{i,j,k,n}} x \qquad (2.5)$$

where, $\overline{S}_0^{yzt}$ is the zeroth order expansion of the pseudo source term, and c_1 and c_2 are constants of integration. The constants of integration can be eliminated using the boundary conditions at the cell surface: $\overline{C}^{yzt}_{i-1,j,k,n} \equiv \overline{C}^{yzt}_{i,j,k,n}(x = -a_i)$ and $\overline{C}^{yzt}_{i,j,k,n} \equiv \overline{C}^{yzt}_{i,j,k,n}(x = +a_i)$, where $\overline{C}^{yzt}_{i-1,j,k,n}$ and $\overline{C}^{yzt}_{i,j,k,n}$ are the cell surface averaged variables at $x = -a_i$ and $x = +a_i$, respectively. Note that the boundary conditions for interior cells are not known, and are in fact the discrete unknown variables of the NIM. A set of discrete equations for surface-averaged variables is obtained in terms of the truncated pseudo-sources by imposing continuity of the flux (for second order ODEs) at interfaces., For example, for $\overline{C}^{yzt}_{i,j,k,n}(x)$, continuity of the flux is imposed at the interface between cell

(i, j, k, n) and (i+1, j, k, n). Similar procedure can be applied to obtain discrete equations for $\overline{C}^{zxt}_{i,j,k,n}$ and $\overline{C}^{xyt}_{i,j,k,n}$.

Similarly, averaging over x, y and z in a cell yields an ODE in time as follows:

$$\frac{d\overline{C}^{xyz}(t)}{dt} = \overline{S}^{xyt}(t) \tag{2.6}$$

where,

$$\overline{S}^{xyz}(t) \equiv \frac{1}{8a_i b_j c_k}\int_{-a_i}^{+a_i}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(\begin{array}{l}\overline{D}\left(-\dfrac{\partial^2 C}{\partial y^2}-\dfrac{\partial^2 C}{\partial z^2}\ -\dfrac{\partial^2 C}{\partial x^2}\right)\\ +\overline{v}\dfrac{\partial C}{\partial y}+\overline{w}\dfrac{\partial C}{\partial z}-s(x,y,z,t)\end{array}\right)dxdydz \tag{2.7}$$

Eqs. (2.6) and (2.7) are valid for all nodes.

The solution of the ODE in the cell (i, j, k, n) after expansion and truncation of the pseudo source term at zeroth order yields,

$$\overline{C}^{xyz}_{i,j,k,n}(t) = c_3 + \overline{S}^{xyz}_{0i,j,k,n}\,t \tag{2.8}$$

Applying the initial condition, $\overline{C}^{xyz}_{i,j,k,n-1} \equiv \overline{C}^{xyz}_{i,j,k,n}(t=-\tau_n)$, yields the value of c_3. Substituting c_3 in Eq. (2.8) and defining $\overline{C}^{xyz}_{i,j,k,n} \equiv \overline{C}^{xyz}_{i,j,k,n}(t=+\tau_n)$ yields a scheme for marching in time. However, just like the three difference schemes obtained by solving the space-dependent ODEs, this scheme is also written in terms of the pseudo source term. Note that these pseudo source terms are unknown.

In the final step, the truncated pseudo-source terms are eliminated by imposing certain constraints. The constraint equations are obtained by satisfying the PDE over a cell in an average sense, i.e.

$$\frac{1}{16a_ib_jc_k\tau_n}\int_{-a_i}^{+a_i}\int_{-\tau_n}^{+\tau_n}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(\begin{array}{l}\dfrac{\partial C}{\partial t}+\bar{u}\dfrac{\partial C}{\partial x}\\ +\bar{v}\dfrac{\partial C}{\partial y}+\bar{w}\dfrac{\partial C}{\partial z}\end{array}\right)dxdydzdt$$
$$=\frac{1}{16a_ib_jc_k\tau_n}\int_{-a_i}^{+a_i}\int_{-\tau_n}^{+\tau_n}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(\begin{array}{l}\bar{D}(\dfrac{\partial^2 C}{\partial x^2}+\dfrac{\partial^2 C}{\partial y^2}+\dfrac{\partial^2 C}{\partial z^2}\ )\\ +s(x,y,z,t)\end{array}\right)dxdydzdt \tag{2.9}$$

and imposing the condition that cell-averaged variables be unique, i.e. independent of the order of integration, which results in the following three equations:

$$\frac{1}{2a_i}\int_{-a_i}^{+a_i}\bar{C}^{yzt}(x)dx=\frac{1}{2b_j}\int_{-b_j}^{+b_j}\bar{C}^{zxt}(y)dy=\frac{1}{2c_k}\int_{-c_k}^{+c_k}\bar{C}^{xyt}(z)dz=\frac{1}{2\tau_n}\int_{-\tau_n}^{+\tau_n}\bar{C}^{xyz}(t)dt \tag{2.10}$$

The NIM procedure as discussed above after a lot of algebra carried out using Mathematica® yields the following set of discrete equations,

$$F_{57}\bar{C}^{yzt}_{i,j,k,n}-F_{51}\bar{C}^{yzt}_{i-1,j,k,n}-F_{52}\bar{C}^{yzt}_{i+1,j,k,n}=F_{53}(\bar{C}^{xyz}_{i,j,k,n}+\bar{C}^{xyz}_{i,j,k,n-1})+F_{54}(\bar{C}^{xyz}_{i+1,j,k,n}+\bar{C}^{xyz}_{i+1,j,k,n-1}) \tag{2.11}$$

$$F_{67}\bar{C}^{zxt}_{i,j,k,n}-F_{61}\bar{C}^{zxt}_{i,j-1,k,n}-F_{62}\bar{C}^{zxt}_{i,j+1,k,n}=F_{63}(\bar{C}^{xyz}_{i,j,k,n}+\bar{C}^{xyz}_{i,j,k,n-1})+F_{64}(\bar{C}^{xyz}_{i,j+1,k,n}+\bar{C}^{xyz}_{i,j+1,k,n-1}) \tag{2.12}$$

$$F_{47}\bar{C}^{xyt}_{i,j,k,n}-F_{41}\bar{C}^{xyt}_{i,j,k-1,n}-F_{42}\bar{C}^{xyt}_{i,j,k+1,n}=F_{43}(\bar{C}^{xyz}_{i,j,k,n}+\bar{C}^{xyz}_{i,j,k,n-1})+F_{44}(\bar{C}^{xyz}_{i,j,k+1,n}+\bar{C}^{xyz}_{i,j,k+1,n-1}) \tag{2.13}$$

$$F_{77}\bar{C}^{xyz}_{i,j,k,n}-F_{78}\bar{C}^{xyz}_{i,j,k,n-1}=f+F_{71}\bar{C}^{xyt}_{i,j,k,n}+F_{72}\bar{C}^{xyt}_{i,j,k-1,n}+F_{73}\bar{C}^{yzt}_{i,j,k,n}+F_{74}\bar{C}^{yzt}_{i-1,j,k,n}+F_{75}\bar{C}^{zxt}_{i,j,k,n}+F_{76}\bar{C}^{zxt}_{i,j-1,k,n}+F_{78}\bar{C}^{xyz}_{i,j,k,n-1} \tag{2.14}$$

In the above equations, the subscripts *i, j, k* and *n* in the coefficients *F* and in *f* are omitted for the sake of clarity. (The subscripts on *F* may seem strange. These are however the last set of coefficients in a string of coefficients defined and used in the development of the final set of equations. The other coefficients were successfully

eliminated in favor of this final set. In order to keep the link between the computer code and the dissertation, it was decided to retain the numbering of these coefficients in the dissertation as they exist in the computer code.)

With the goals of developing a domain decomposition based parallel scheme for this 3D, time-dependent nodal approach in mind, it is important to know the dependency-template of each and every coefficient and parameter; i.e. the neighboring nodes on which evaluation of a certain coefficient depends must be clearly identified. This information is important for data transfer needs between neighboring processors. The coefficients *F* at the cell (*i, j, k*) in the equation for $\bar{C}^{yzt}$ (Eq. 2.11) are functions of *u*-velocity and node dimension a_i of the node as well as the same quantities in the neighboring node in the positive *x*-direction. These coefficients in the equation for $\bar{C}^{zxt}$ (Eq. 2.12) are functions of *v*- velocity and node dimension *b* of the node as well as the same quantities in the neighboring node in the positive *y*-direction. Similar dependence exists in the equation for $\bar{C}^{xyt}$. However, in the equation for $\bar{C}^{xyz}$ (Eq. 2.14), the *F* coefficients depend on all three components of the velocities as well as all three dimensions of the node. Also, the coefficient *f* in the equation is the source term averaged over the node i.e.,

$$f_{i,j,k,n} = \frac{1}{16 a_i b_j c_k \tau_n} \int_{-\tau_n}^{\tau_n} \int_{-c_k}^{+c_k} \int_{-b_j}^{+b_j} \int_{-a_i}^{+a_i} S(x, y, z, t)\, dx dy dz dt \qquad (2.15)$$

The location of various derived variables is shown in Fig. 2.1. The variable $\bar{C}^{xyz}_{i,j,k,n}$ is connected to the same variable at the previous time step ($\bar{C}^{xyz}_{i,j,k,n-1}$) through Eq. (2.14). This variable can be considered to be the surface averaged variable over a surface parallel

to the *xyz*-plane in a four dimensional space-time continuum, as has been suggested in previous works. However, it is more convenient to describe the variable separately especially for the development of the parallelization scheme. It should be noted that the description of the variable, or how it is interpreted, is immaterial to the development of parallel scheme.

2.2 Boundary Conditions for the Convection Diffusion Equation

2.2 1 Dirichlet Boundary Condition

The implementation of Dirichlet boundary conditions is straightforward in MNIM formulation. For example, in a cell (*i*, *j*, *k*) which has its surface at $x = -a_i$ (in the cell centered local coordinate system) at the boundary, the boundary condition is implemented in the following manner. In the Eq. (2.11), the variable $\bar{C}^{yzt}_{i-1,j,k,n}$ is pre-assigned a value obtained by averaging the variable *C* over the abovementioned surface. If the variable is varying with time, it is averaged in time, in addition to the averaging over the surface discussed above. Based on the above discussion, the method of implementation of the Dirichlet boundary condition on any other surface is obvious.

2.2 2 Robin Boundary Condition

In the work carried out by Wang (2003), only Dirichlet boundary conditions were discussed. The implementation of the Robin (or mixed) boundary condition was

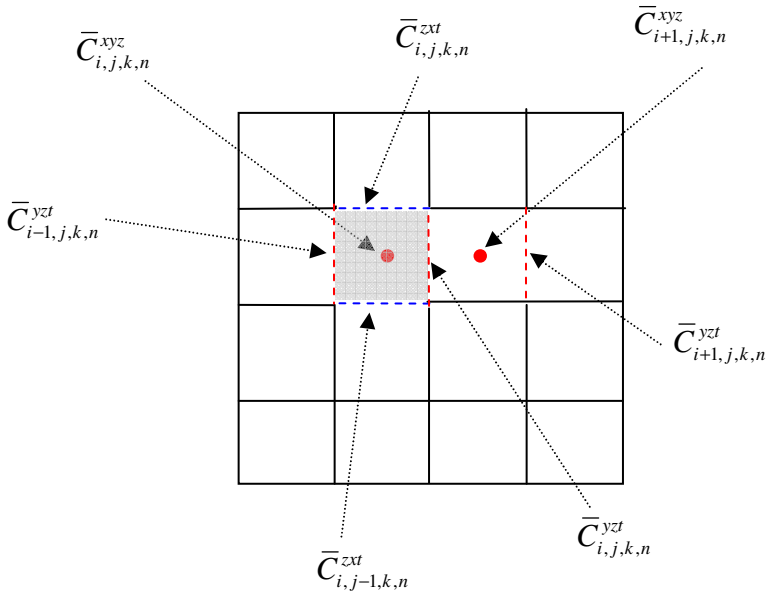

Fig. 2.1 Location of surface-averaged variables corresponding to variable C. The shaded cell in the figure is cell (i, j, k). Only x-y plane cutting through the vertical (z-direction) center of the cells has been shown for the sake of clarity. Variables $\overline{C}^{xyt}_{i,j,k-1,n}$ and $\overline{C}^{xyt}_{i,j,k,n}$ (not shown here) are located at the bottom and top surfaces of the node (i, j, k), respectively.

developed by Michael (2001) earlier in the context of convection-diffusion equation. However, the present formulation is developed in terms of *F* coefficients so as to make its implementation in a computer code relatively straightforward.

The mixed boundary condition at $x=-a_i$ (in local cell coordinates can be written as:

$$\alpha(y,z,t)\frac{\partial C}{\partial x}(x=-a_i,y,z,t)+\beta(y,z,t)C(x=-a_i,y,z,t) = \gamma(y,z,t) \tag{2.16}$$

where α,β and γ are known parameters, Averaging over *y*, *z* and *t* in a cell, one gets,

$$\bar{\alpha}^{yzt}_{j,k,n}\frac{\partial \bar{C}^{yzt}_{i,j,k,n}}{\partial x}(x=-a_i)+\bar{\beta}^{yzt}_{j,k}\bar{C}^{yzt}_{i,j,k,n}(x=-a_i)=\bar{\gamma}^{yzt}_{j,k,n} \tag{2.17}$$

Also, average of products is approximated as product of averages which is consistent with the order of the scheme as discussed earlier. The solution for the ODE for $\bar{C}^{yzt}_{i,j,k,n}$, given in Eq. (2.5) is substituted in Eq. (2.17) to obtain an equation for $\bar{C}^{yzt}_{i-1,j,k,n}$. After substituting values of pseudo source terms and constants as obtained for Eq. (2.5), final equation of $\bar{C}^{yzt}_{i-1,j,k,n}$ is written as:

$$\left(\bar{\alpha}^{yzt}_{j,k,n}(F_{52i-1,j,k}+F_{54i-1,j,k})+\bar{\beta}^{yzt}_{j,k,n}\right)\bar{C}^{yzt}_{i-1,j,k,n}-\bar{\alpha}^{yzt}_{j,k,n}F_{52i-1,j,k}\bar{C}^{yzt}_{i,j,k,n} = F_{54i-1,j,k}(\bar{C}^{xyz}_{i,j,k,n}+\bar{C}^{xyz}_{i,j,k,n-1})+\bar{\gamma}^{yzt}_{j,k,n} \tag{2.18}$$

The Neumann boundary conditions can be easily obtained from the above equation by setting $\beta=0$. The symmetry boundary condition, which is a Neumann boundary

condition, can be obviously implemented through the above equation. The boundary conditions on the other surfaces can be implemented analogously.

2.3 MNIM for the Navier-Stokes Equations

In the early applications of NIM, non-linear terms as well as pressure terms were merged with the pseudo source-terms. This approach however was not very satisfactory. An alternative approach used was to write the Navier-Stokes equation valid over a cell as follows:

$$\begin{aligned}&\frac{\partial u(x,y,z,t)}{\partial t}+u_0\frac{\partial u(x,y,z,t)}{\partial x}+v_0\frac{\partial u(x,y,z,t)}{\partial y}+w_0\frac{\partial u(x,y,z,t)}{\partial z}\\&=\nu\left(\frac{\partial^2 u(x,y,z,t)}{\partial x^2}+\frac{\partial^2 u(x,y,z,t)}{\partial y^2}+\frac{\partial^2 u(x,y,z,t)}{\partial z^2}\right)+s(x,y,z,t)\end{aligned} \tag{2.19}$$

where, u_0, v_0, and w_0 are cell averaged velocities, and

$$s(x,y,z,t)=-\frac{1}{\rho}\frac{\partial p}{\partial x}+g_x \tag{2.20}$$

Similar, momentum equations can be written for the y and z directions. With the velocities in the convection term approximated as in Eq. (2.19), the momentum equations can be treated similarly to the convection-diffusion equation. The f term for the x – momentum equation corresponding to the nodal scheme for the convection diffusion equation is given by,

$$f_{i,j,k,n}=\frac{1}{16a_ib_jc_k\tau_n}\int_{-\tau_n}^{\tau_n}\int_{-c_k}^{+c_k}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(-\frac{1}{\rho}\frac{\partial p}{\partial x}+g_x\right)dxdydzdt \tag{2.21}$$

The pressure term is evaluated as follows,

$$\frac{1}{16a_i b_j c_k \tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-c_k}^{+c_k}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(-\frac{1}{\rho}\frac{\partial p}{\partial x}\right)dxdydzdt$$
$$=\frac{1}{2a_i}\int_{-a_i}^{+a_i}\left(-\frac{1}{\rho}\frac{d\overline{p}^{yzt}}{dx}\right)dx \tag{2.22}$$

This can be integrated to obtain,

$$\int_{-a_i}^{+a_i}\left(-\frac{1}{\rho}\frac{d\overline{p}^{yzt}}{dx}\right)dx=\overline{p}_{i,j,k}^{yzt}-\overline{p}_{i-1,j,k}^{yzt} \tag{2.23}$$

Similar steps are needed to evaluate the pressure terms in the *y* and *z*- direction momentum equations.

Equation (2.19) is similar to Eq. (2.1) and can be solved in a similar manner. However, the *F* coefficients depend on the cell-averaged velocities; therefore the cell averaged velocities must also be updated during the iteration steps Unlike the convection-diffusion equation in which the velocities and hence the coefficients *F* were fixed, for the N-S equations the *F* coefficients need to be reevaluated at each iteration.

The above formulation requires evaluation of the *F* coefficients at each iteration. The *F* coefficients are quite complex functions involving exponential and linear terms. Therefore, the reevaluation of these coefficients for each iteration results in a heavy computational burden. In order to avoid such reevaluations the scheme is modified as follows. The momentum equation in a node is written as

$$\frac{\partial u(x,y,z,t)}{\partial t}+\overline{u}_p\frac{\partial u(x,y,z,t)}{\partial x}+\overline{v}_p\frac{\partial u(x,y,z,t)}{\partial y}+\overline{w}_p\frac{\partial u(x,y,z,t)}{\partial z}$$
$$=\nu\left(\frac{\partial^2 u(x,y,z,t)}{\partial x^2}+\frac{\partial^2 u(x,y,z,t)}{\partial y^2}+\frac{\partial^2 u(x,y,z,t)}{\partial z^2}\right)+s(x,y,z,t) \tag{2.24}$$

where,

$$s(x,y,z,t) = -\frac{1}{\rho}\frac{\partial p}{\partial x} + (\overline{u}_0 - \overline{u}_p)\frac{\partial u}{\partial x} + (\overline{v}_0 - \overline{v}_p)\frac{\partial u}{\partial y} + (\overline{w}_0 - \overline{w}_p)\frac{\partial u}{\partial z} + g_x \qquad (2.25)$$

and $\overline{u}_0$, $\overline{v}_0$, $\overline{w}_0$ and $\overline{u}_p$, $\overline{v}_p$, $\overline{w}_p$ are cell averaged velocities at current time step and previous time step, respectively. The momentum equations in the *y* and *z* – directions can be written similarly. The rest of the procedure is then similar to the convection diffusion equation. The *f* terms in the MNIM are now evaluated in the same way as described for the pressure term in the previous formulation (see equations (2.21), (2.22) and (2.23)). The resulting expression for *f* term is given as:

$$\begin{aligned} f_{i,j,k,n} = {} & \frac{\left(\overline{p}^{yzt}_{i,j,k} - \overline{p}^{yzt}_{i-1,j,k}\right)}{2a_i} + (\overline{u}_0 - \overline{u}_p)\frac{(\overline{u}^{yzt}_{i,j,k,n} - \overline{u}^{yzt}_{i-1,j,k,n})}{2a_i} \\ & + (\overline{v}_0 - \overline{v}_p)\frac{(\overline{u}^{zxt}_{i,j,k,n} - \overline{u}^{zxt}_{i,j-1,k,n})}{2b_j} + (\overline{w}_0 - \overline{w}_p)\frac{(\overline{u}^{xyt}_{i,j,k,n} - \overline{u}^{xyt}_{i,j,k-1,n})}{2c_k} + \overline{g}^{xyzt}_x \end{aligned} \qquad (2.26)$$

In the above expression subscript (*i*, *j*, *k*, *n*) has been omitted from $\overline{u}_0$, $\overline{u}_p$, etc. for the sake of brevity. Similar expressions can be written for *v* and *w* momentum equations.

The implementation of the boundary conditions for momentum equations is same as that for convection-diffusion equation.

2.4 MNIM for Pressure Poisson Equation

In addition to the momentum equations, in the MNIM a pressure Poisson equation is solved in place of the continuity equation (Wang 2003), which is given by

$$\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}+\frac{\partial^2 p}{\partial z^2}=s_p(x,y,z,t) \tag{2.27}$$

where $s_p(x,y,z,t)$ is given as,

$$s_p(x,y,z,t)=-\sum_{i=1}^{3}\sum_{j=1}^{3}\left(\frac{\partial u_i}{\partial x_j}\right)\left(\frac{\partial u_j}{\partial x_i}\right)+\frac{\partial g_x}{\partial x}+\frac{\partial g_y}{\partial y}+\frac{\partial g_z}{\partial z} \tag{2.28}$$

In the above equation u_1, u_2, u_3, x_1, x_2 and x_3 are u, v, w, x, y and z, respectively. The development of the scheme is similar to the scheme developed for the momentum equations and is discussed briefly. Applying the transverse integration procedure on Eq. (2.28) to obtain the ODE in the x –direction, one gets,

$$\frac{d^2\overline{p}^{yzt}}{dx^2}=\overline{S}_{p0} \tag{2.29}$$

The above ODE is written with truncated pseudo source term. The ODE is solved to obtain a solution with quadratic and linear terms. ODEs in the other two directions can be obtained similarly. The rest of the procedure, which is similar to the procedure for the momentum equations, leads to the following set of linear equations:

$$\begin{aligned}F_{27}\overline{p}^{yzt}_{i,j,k,n}-F_{21}\overline{p}^{yzt}_{i-1,j,k,n}-F_{22}\overline{p}^{yzt}_{i+1,j,k,n}=F_{23}(\overline{p}^{zxt}_{i,j,k,n}+\overline{p}^{zxt}_{i,j-1,k,n})+\\ F_{24}(\overline{p}^{zxt}_{i+1,j,k,n}+\overline{p}^{zxt}_{i+1,j-1,k,n})+F_{25}(\overline{p}^{xyt}_{i,j,k,n}+\overline{p}^{xyt}_{i,j,k-1,n})+\\ F_{26}(\overline{p}^{xyt}_{i+1,j,k,n}+\overline{p}^{xyt}_{i+1,j,k-1,n})+F_{28}f_{1i,j,k,n}+F_{29}f_{1i+1,j,k,n}\end{aligned} \tag{2.30}$$

$$\begin{aligned}F_{37}\overline{p}^{zxt}_{i,j,k,n}-F_{31}\overline{p}^{zxt}_{i,j-1,k,n}-F_{32}\overline{p}^{zxt}_{i,j+1,k,n}=F_{33}(\overline{p}^{xyt}_{i,j,k,n}+\overline{p}^{xyt}_{i,j,k-1,n})+\\ F_{34}(\overline{p}^{xyt}_{i,j+1,k,n}+\overline{p}^{xyt}_{i,j+1,k-1,n})+F_{35}(\overline{p}^{yzt}_{i,j,k,n}+\overline{p}^{yzt}_{i-1,j,k,n})+\\ F_{36}(\overline{p}^{yzt}_{i,j+1,k,n}+\overline{p}^{yzt}_{i-1,j+1,k,n})+F_{38}f_{1i,j,k,n}+F_{39}f_{1i,j+1,k,n}\end{aligned} \tag{2.31}$$

$$
\begin{aligned}
F_{47}\overline{p}^{xyt}_{i,j,k,n} - F_{41}\overline{p}^{xyt}_{i,,k-1,n} - F_{42}\overline{p}^{xyt}_{i,k+1,n} = F_{43}(\overline{p}^{yzt}_{i,j,k,n} + \overline{p}^{yzt}_{i,j,k-1,n}) + \\
F_{44}(\overline{p}^{yzt}_{i,j+1,k,n} + \overline{p}^{yzt}_{i,j+1,k-1,n}) + F_{45}(\overline{p}^{zxt}_{i,j,k,n} + \overline{p}^{zxt}_{i-1,j,k,n}) + \\
F_{46}(\overline{p}^{zxt}_{i,j+1,k,n} + \overline{p}^{zxt}_{i-1,j+1,k,n}) + F_{48}f_{1i,j,k,n} + F_{49}f_{1i,j,k+1,n}
\end{aligned}
\tag{2.32}
$$

where $\overline{p}^{yzt}_{i-1,j,k,n} \equiv \overline{p}^{yzt}_{i,j,k,n}(x=-a_i)$ and $\overline{p}^{yzt}_{i,j,k,n} \equiv \overline{p}^{yzt}_{i,j,k,n}(x=+a_i)$. All other variables are similarly defined. The subscripts *i, j, k* and *n* in *F* coefficients and in f_2 are omitted. The *F* coefficients in the pressure equations are dependent only on the node dimensions and hence need to be evaluated only once. The f_1 terms are given as follows:

$$
f_{1i,j,k,n} = \frac{1}{16a_ib_jc_k\tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-a_i}^{+a_i}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k} s_p(x,y,z)\,dxdydzdt \tag{2.33}
$$

The product of the velocity derivatives in the above equation are evaluated using the following approximation:

$$
\begin{aligned}
&\frac{1}{16a_ib_jc_k\tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-a_i}^{+a_i}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}\right)dxdydzdt \\
&\approx \frac{1}{16a_ib_jc_k\tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-a_i}^{+a_i}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(\frac{\partial u}{\partial x}\right)dxdydzdt \\
&\times\frac{1}{16a_ib_jc_k\tau_n}\int_{-\tau_n}^{+\tau_n}\int_{-a_i}^{+a_i}\int_{-b_j}^{+b_j}\int_{-c_k}^{+c_k}\left(\frac{\partial v}{\partial y}\right)dxdydzdt
\end{aligned}
\tag{2.34}
$$

The above approximation, namely approximating the average of a product by the product of the averages is second order accurate and is consistent with the order of the MNIM.

2.4.1 Pressure Boundary Conditions

The Neumann boundary condition for the pressure at the surface defined by $x = -a_i$ is given as,

$$\frac{\partial p}{\partial x}(x = -a_i, y, z, t) = \gamma_p(y, z, t) \tag{2.35}$$

Averaging over *y*, *z* and *t*, yields,

$$\frac{d\overline{p}^{yzt}_{i,j,k,n}}{dx}(x = -a_i) = \overline{\gamma}^{yzt}_{p;j,k,n} \tag{2.36}$$

Applying the process similar to that applied for boundary condition for convection-diffusion equation, one gets,

$$\begin{aligned}(F_{21} + 2F_{23} + 2F_{25})\overline{p}^{yzt}_{i-1,j,k,n} - F_{21}\overline{p}^{yzt}_{i,j,k,n} = \overline{\gamma}^{yzt}_{p;j,k,n} + F_{23}(\overline{p}^{zxt}_{i,j,k,n} + \overline{p}^{zxt}_{i,j-1,k,n}) + \\ + F_{25}(\overline{p}^{xyt}_{i,j,k,n} + \overline{p}^{xyt}_{i,j,k-1,n}) + F_{28}f_{1i,j,k,n}\end{aligned} \tag{2.37}$$

Although, same equation was developed by Wang (2003), the current formulation is obtained in terms of *F* coefficients. The earlier formulation was given in terms of the cell dimensions. The present formulation is more convenient in terms of the implementation in a computer code.

The boundary conditions for the pressure at a wall parallel to the y-z plane are obtained by noting that $u = v = w = \partial u/\partial t = \partial^2 u/\partial y^2 = \partial^2 u/\partial z^2 = 0$, at that wall. Substituting these values in the *v* – momentum equation, one gets,

$$\frac{\partial p}{dx} = \nu \frac{\partial^2 u}{\partial x^2} + g_x \tag{2.38}$$

The wall boundary condition can be obtained by replacing $\overline{\gamma}^{yzt}_{j,k,n}$ by $\nu\, d^2\overline{u}^{yzt}_{i-1,j,k,n}/dx^2 + \overline{g}^{yzt}_{x}$ in Eq. (2.37). In order to maintain the order of the overall scheme,

the second derivative of the velocity in the above expression, fourth order approximation of that derivative is needed. Since fourth order nodal method is quite complicated, the finite difference method is used in its place (Wang 2003).

The implementation of the Dirichlet boundary condition is straightforward and is same as that for convection-diffusion scheme. Only Dirichlet and Neumann boundary conditions for the pressure Poisson equation are used in the present work.

It should be noted that, in the 3-D MNIM, four algebraic equations need to be solved for each velocity component (or temperature) as there are four variables corresponding to each velocity component ($\overline{u}^{yzt}$, $\overline{u}^{zxt}$, $\overline{u}^{xyt}$, $\overline{u}^{xyz}$). Similarly there are three discrete variables for the pressure ($\overline{p}^{yzt}$, $\overline{p}^{zxt}$, $\overline{p}^{xyt}$).

Chapter 3

Modified Nodal Integral Method for Turbulent Flows (k-ε Model)

In general, the goal of the turbulence modeling is to obtain the average of various quantities of interest such as velocity and pressure in a fluctuating turbulent flow. Reynolds averaged Navier Stokes (RANS) equations, for the average quantities, are obtained by averaging the Navier Stokes equations. Since RANS equations lack closure, the Reynolds stresses appearing in the equations are modeled in terms of the eddy viscosity, which is then approximated. Several models have been proposed through the last few decades for the approximation of the eddy viscosity. The classical models are classified as zero equation models (Smith 1967, Baldwin 1978), one equation models (e.g. Spalart 1994, Baldwin 1990), and the two equation models (e.g. k-ε, k-ω model (Wilcox 1988)). In the Reynolds stresses model (e.g. Launder 1975), the six equations for Reynolds stresses are directly solved instead of defining an eddy viscosity.

The k-ε model is based on the postulates given by Harlow (1967). The most commonly used k-ε model, known as the standard k-ε model, was developed by Jones (1972). Several variations of this model were developed mainly to deal with the near wall characteristics of turbulence. Further discussion of these models is given later in this

chapter. Some *k-ε* models are derived using the renormalization group theory (RNG *k-*ε models, e.g. Yakhot (1992)). However, these models are quite similar to the standard *k-ε* model, and in general do not yield results significantly different from the standard *k-ε* model. The implementation of the RNG based models is also similar to that for the standard *k-ε* model.

In this chapter, development and implementation of MNIM for the simulation of the turbulent flows using the *k-ε* model is discussed. Two variations of the model, one from the category of high Reynolds number and another from the category low Reynolds number are used for the implementation. The details of these variations of the model and their implementation are discussed in the subsequent sections. Simulations are carried out for some simple flows in order to validate the scheme.

3.1 Reynolds Averaged Navier-Stokes Equations

The Reynolds Averaged Navier-Stokes (RANS) equations are the ensemble (or time averaged in the case of stationary flows) averaged Navier-Stokes equations. The velocities in turbulent flows fluctuate around a mean value. The RANS equations are obtained by using the fact that mean value of these fluctuations is zero. The RANS equations for the incompressible Newtonian fluid are given as:

$$\frac{\partial \overline{u}_m}{\partial x_m} = 0 \tag{3.1}$$

$$\frac{\partial \overline{u}_m}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_m}{\partial x_p} = -\frac{1}{\rho}\frac{\partial p}{\partial x_m} + \nu \frac{\partial^2 \overline{u}_m}{\partial x_p \partial x_p} - \frac{\partial}{\partial x_p}\overline{u'_m u'_p} + b_m . \tag{3.2}$$

In the above equation, and in the subsequent equations in this chapter, tensor notation is used and Einstein's summation convention is employed. It is obvious that in the absence

of an equation for $\overline{u_m' u_p'}$ the system of equation given above is not closed. Although an equation for $\overline{u_m' u_p'}$ can be developed, it consists of terms which need further equations for the closure of the set of equations. In fact, the RANS equations do not have a closure. Therefore, $\overline{u_m' u_p'}$ is approximated, usually by eddy viscosity model which is given as:

$$\overline{u_m' u_p'} = -\nu_t \left(\frac{\partial \overline{u}_m}{\partial x_p} + \frac{\partial \overline{u}_p}{\partial x_m} \right) \tag{3.3}$$

which results in the following form of Eq. (3.2):

$$\frac{\partial \overline{u}_m}{\partial t} + \overline{u}_p \frac{\partial \overline{u}_m}{\partial x_p} = -\frac{1}{\rho} \frac{\partial p}{\partial x_m} + \frac{\partial}{\partial x_p} \left(\nu + \nu_t \right) \frac{\partial \overline{u}_m}{\partial x_p} + b_m . \tag{3.4}$$

The eddy viscosity (ν_t) needs to be approximated for solution of the RANS equations. The approximation of the eddy viscosity can be accomplished using the *k*-ε model.

3.2 *k*-ε Model

In the *k*-ε equations, *k* is the turbulent kinetic energy given as:

$$k = \frac{1}{2} \overline{u_m' u_m'} \tag{3.5}$$

and ε is the dissipation of *k* given by,

$$\varepsilon = 2\nu s_{mp}' s_{mp}' \tag{3.6}$$

where

$$s_{mp}' = \left(\frac{\partial u_m'}{\partial x_p} + \frac{\partial u_p'}{\partial x_m} \right). \tag{3.7}$$

The conservation equation for k is obtained by writing the equation for $\overline{u_i' u_j'}$ and then contracting i with j. In addition several terms have been approximated to get the following equation for k,

$$\frac{\partial k}{\partial t} + \overline{u}_p \frac{\partial k}{\partial x_p} = \frac{\partial}{\partial x_p}\left(\nu + \frac{\nu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_p} + 2\nu_t \overline{s}_{mp}\overline{s}_{mp} - \varepsilon \tag{3.8}$$

where,

$$\overline{s}_{mp} = \left(\frac{\partial \overline{u}_m}{\partial x_p} + \frac{\partial \overline{u}_p}{\partial x_m}\right). \tag{3.9}$$

The equation for ε is given as:

$$\frac{\partial \varepsilon}{\partial t} + \overline{u}_p \frac{\partial \varepsilon}{\partial x_p} = \frac{\partial}{\partial x_p}\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_p} + 2C_{1\varepsilon}\frac{\varepsilon}{k}\nu_t \overline{s}_{mp}\overline{s}_{mp} - C_{2\varepsilon}\frac{\varepsilon^2}{k}. \tag{3.10}$$

The eddy viscosity in terms of k and ε is defined as,

$$\nu_t = C_\mu \frac{k^2}{\varepsilon}. \tag{3.11}$$

The set of Eqs. (3.1), (3.4), (3.8), (3.10) and (3.11) is known as the standard k-ε model. The model constants are usually assigned the values given in Table 3.1.

$C_{1\varepsilon}$	$C_{2\varepsilon}$	C_μ	σ_ε	σ_k
1.44	1.92	0.09	1.2	1.0

Table 3.1 Typical values of constants in the standard k-ε model

It should be noted that there are several variations of the k-ε model, and these variations use different values for these and/or other constants. Moreover, the equations in various k-ε models may also vary slightly. Some of these variations are discussed later in this chapter. The boundary conditions for the different models used in the present work are also discussed.

3.3 Near Wall k-ε Modeling

The modeling approach discussed so far is not valid in the regions near the walls because the model constants are not calibrated for these regions. Moreover, flow becomes laminar in the vicinity of the wall and therefore, eddy viscosity should go to zero there. Several approaches have been developed to overcome this limitation of the k-ε model. These approaches are broadly classified as: (a) low Reynolds number models; and (b) high Reynolds number models. A low Reynolds number model, as the name suggests, is valid for the flows with relatively low Reynolds number. In these flows the effect of the wall is present in the regions which are significantly away from the wall. However, for the high Reynolds number flows the effect of the wall is present only in the immediate vicinity of the wall, hence the need for a different modeling strategy.

3.3.1 Low Reynolds Number k-ε Model

In the low Reynolds number formulation, the Eqs. (3.8), (3.9) and (3.11) are modified to incorporate the effect of the wall. The low Reynolds model correctly predicts the flow in the viscous layer, next to the wall. These modifications are usually implemented in such a way that model reduces to its standard form in the regions away from the wall. Several variations of low Reynolds model, for example those given by Chen (1982), Nagano (1990), and Rodi(1993), are available in the literature. The model implemented here was proposed by Abe, Kondoh and Nagano (1993) (hereafter referred to as AKN model). It should be noted that the implementation method will not be significantly different for other variations of the low Reynolds number model.

The AKN model is developed by multiplying constant C_μ in Eq. (3.11) by f_μ and multiplying $C_{2\varepsilon}$ in Eq. (3.10) by g_ε. The f_μ is defined as follows:

$$f_\mu \equiv \left\{1-\exp\left(-\frac{y^*}{14}\right)\right\}^2\left[1+\frac{5}{R_t^{3/4}}\exp\left(1-\frac{R_t}{200}\right)^2\right] \tag{3.12}$$

and g_ε is defined as :

$$g_\varepsilon \equiv \left\{1-\exp\left(-\frac{y^*}{3.1}\right)\right\}^2\left[1-0.3\exp\left(1-\frac{R_t}{6.5}\right)^2\right]. \tag{3.13}$$

In the above equations y^* is defined as follows:

$$y^* \equiv \frac{u_\varepsilon d}{\nu} \tag{3.14}$$

where *d* is the normal distance of the node center from the wall and $u_\varepsilon = (\nu\varepsilon)^{1/4}$. In Eqs. (3.12) and (3.13) R_t is defined as follows:

$$R_t \equiv \frac{k^2}{\nu\varepsilon}. \tag{3.15}$$

In addition to modifying the equations, some of the model constants are also changed in the AKN model from those used in the standard *k*-ε model. The constants used for this model are given in Table 3.2.

$C_{1\varepsilon}$	$C_{2\varepsilon}$	C_μ	σ_ε	σ_k
1.5	1.9	0.09	1.4	1.4

Table 3.2 Values of constants used in the AKN low Reynolds number model.

In addition to the boundary condition on the velocities, boundary conditions for the *k* and ε are also needed. Since no slip and no penetration condition are applied at the walls, the *k* goes to zero at the walls. The boundary condition for *ε*, can be derived from

Eq. (3.8). At the wall, k, ν_t and the velocity are zero. Moreover, derivatives of velocity in the direction parallel to the wall are also zero. Therefore, Eq. (3.8) reduces to the following:

$$\varepsilon_{wall} = \nu \frac{\partial^2 k}{\partial n^2} \tag{3.16}$$

where n is the wall normal direction. The above equation is used as the boundary condition for ε.

3.3.2 High Reynolds Number k-ε Model

Although, the AKN model given above is in general valid for any Reynolds number, extremely fine grid is needed to resolve thin viscous layers formed in the case of high Reynolds number flows. For this reason it is often used only for low Reynolds number flows. To avoid increased computational load because of the need for a much finer grid, a different strategy is used for the high Reynolds number flows.

It is well known that in flows parallel to a wall (flow over a flat plate, channel flow or pipe flow) there is a region where velocity follows log-law. In the channel and pipe flow the region lies between distances, $d^+ = 30$ and $d/\delta = 30$, from the wall, where δ is the half-width of the channel (or radius of the pipe), and the distance d^+ is in the wall units and is related to the actual distance d as follows:

$$d^+ \equiv \frac{d u_\tau}{\nu} \tag{3.17}$$

where u_τ is defined as:

$$u_\tau \equiv \sqrt{\frac{\tau_{wall}}{\rho}} \tag{3.18}$$

Here, τ_{wall} is the shear stress at the wall.

In the high Reynolds number model, it is assumed that the flow is predominantly parallel to the wall in its vicinity. The assumption can be considered valid in many cases because the flow is essentially parallel to the walls in the boundary layers. However, the validity of log-law cannot be assumed in the regions where flow is not parallel to one of the walls, as is the case for flow near the edge of the two walls. Nevertheless, the model is commonly used for all the flows because of its relative simplicity.

Two cells near a wall are shown in the Fig. 3.1. In the high Reynolds number model, the cells near the wall are chosen in such a way that the interface shown in the figure lies in the log-law region. The standard *k*-*ε* equations are solved for all the cells except for the cells adjacent to the wall. At the interface shown in the Fig. 3.1, the friction velocity (u_τ) is evaluated from the following equation

$$\frac{U_1 - U_{wall}}{u_\tau} = \frac{1}{\kappa} \log y_1^+ + 5.5 \tag{3.19}$$

where, y_1^+ is the distance of the abovementioned interface from the wall in the wall units, $U_1 - U_{wall}$ is the velocity of the fluid at that interface relative to the wall, and κ is the von Karman constant and has been assigned a value of 0.41. The *k* and *ε* equations are not solved in the cell next to wall. Instead, the following boundary conditions for these equations are applied at the interface shown in the Fig. (3.1):

$$k = \frac{u_\tau^2}{\sqrt{C_\mu}} \tag{3.20}$$

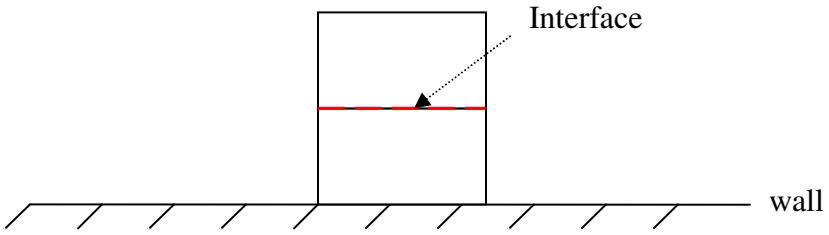

Fig. 3.1Two cells next to the wall. The interface shown by the dashed (red) line should lie in the region where log-law is valid.

$$\varepsilon = \frac{u_\tau^3}{n\sqrt{C_\mu}} \tag{3.21}$$

where n is the distance of the abovementioned interface from the wall.

The U_1 in Eq. (3.19) is obtained by solving the momentum equation for the velocity component parallel to the wall. The flow in the near wall region is laminar and hence the convection terms are ignored during the solution of the equation. Also, the wall normal velocity is made zero at the interface discussed above. The cells on the edges between two surfaces need to be treated differently because there is one velocity component parallel to one of the wall and another component parallel to the other wall. Therefore, both of these velocity components need to be evaluated. The above method for solving for U_1, which is used in the present work, is given by Nonino (1988). Alternatively, equation for U_1 can also be derived by using $\nu \partial U/\partial n = \tau_{wall}$ where n is wall normal direction and τ_{wall} can be obtained from Eq. (3.18) (Durbin 2001).

3.4 Implementation of *k-ε* Model using MNIM

In chapter 2, the MNIM for the Navier-Stokes and convection-diffusion equation was discussed. Though, the scheme developed is valid for spatially varying dynamic viscosity (or diffusion coefficient), it has been previously developed only for constant properties. Since, the eddy viscosity is a variable quantity; the Navier-Stokes equation *in a cell* is approximated as follows:

$$\frac{\partial \overline{u}_m}{\partial t} + \overline{u}_p \frac{\partial \overline{u}_m}{\partial x_p} = -\frac{1}{\rho}\frac{\partial p}{\partial x_m} + \left(\nu + \nu_{t0}\right)\frac{\partial^2 \overline{u}_m}{\partial x_p \partial x_p} + b_m \tag{3.22}$$

where, ν_{t0} in a cell at the current time step is defined in terms of the cell averaged k and ε, as follows:

$$\nu_{t0i,j,k,n} = C_\mu \frac{\left(\overline{k}^{xyz}_{i,j,k,n}\right)^2}{\overline{\varepsilon}^{xyz}_{i,j,k,n}} \tag{3.23}$$

Note that the above value of the eddy viscosity is an approximation of the cell averaged eddy viscosity. In this approximation the average of products is approximated by the product of the averages. The approximation is second order accurate as has been discussed previously.

Moreover, unlike the constant property case, the derivatives at the cell interfaces are not continuous; instead the product of diffusion coefficient and the derivative is continuous. The modified F coefficients (see Eqs (2.11)-(2.14)) required for the Navier-Stokes, k and ε equations can be evaluated based on the diffusion coefficient in the respective equation and considering the modified equation for the continuity of flux. The f coefficient (see Eq. (2.14)), which is obtained by averaging the source term over the cell, remains the same for the Navier-Stokes equations.

Retaining the convection, diffusion and time derivative terms on the left hand side and treating all the other terms in the equations of k and ε as source terms, the development follows in the manner similar to that for the generic convection-diffusion equation discussed in the Chapter 2. Since the diffusion coefficients for these equations also vary in space, they can hence be treated in the same way as discussed above for the Navier-Stokes equation. Obviously, the F coefficients for these equations are different from those for the Navier-Stokes equation and need to be evaluated separately for each of

these equations. Considering the source terms, the *f* coefficient, which depend only on the source terms, needs to be evaluated for each of these equations.

The *f* coefficient for the ε equation (called f_ε) is evaluated as follows:

$$f_\varepsilon = 2C_{1\varepsilon}\frac{\varepsilon_0}{k_0}\nu_{t0}\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0 - C_{2\varepsilon}\frac{\varepsilon_0^2}{k_0} \tag{3.24}$$

Ideally, the k_0 and ε_0 for a cell in the above equation should be obtained by averaging *k* and ε, respectively, over space and time in that cell. However, since in the present work only steady state solution is considered, the quantities averaged only over space are used instead (e.g. $\varepsilon_{0i,j,k,n} \approx \overline{\varepsilon}^{xyz}_{i,j,k,n}$). This may results in *false transients*; however, the steady state solutions will not be affected. The product of various derivatives of velocities with respect to *x*, *y* and *z* appear in the equation for $\overline{s}_{mp}\overline{s}_{mp}$. The $\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0$ is the time-space average of $\overline{s}_{mp}\overline{s}_{mp}$ over a cell and can be evaluated in the same fashion as other terms of similar type are evaluated in Eq. (2.34). It should be noted that the average of products is approximated again by product of averages in the evaluation of f_ε.

Similarly, f_k is obtained as follows:

$$f_k = 2\nu_{t0}\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0 - \varepsilon_0 \tag{3.25}$$

Values on the right hand side of Eq. (3.25) are evaluated in the same way as in equation (3.24).

Details of the implementation of the low Reynolds number and high Reynolds number models in the MNIM code are discussed in the next two sub-sections.

3.4.1 Implementation of Low Reynolds Number *k-ε* Model using MNIM

In the implementation of the low Reynolds number model, the expressions in Eqs. (3.12) to (3.15) are evaluated by replacing k and ε by $\overline{k}_{i,j,k,n}^{xyz}$ and $\overline{\varepsilon}_{i,j,k,n}^{xyz}$, respectively. The distance d in Eq. (3.14) is considered to be the distance of the cell center from the wall.

One problem commonly found in low Reynolds number models is that values of k and/or ε can become negative during the false transients (or during iterations) which commonly leads to iterations that do not converge. To overcome this problem, the discretized equation corresponding to the k equation (Eq. (3.8)) needs to be modified. The abovementioned discretized equation (see Eq. (2.14)) is given as:

$$F_{77}\overline{k}_{i,j,k,n}^{xyz} - F_{78}\overline{k}_{i,j,k,n-1}^{xyz} = 2\nu_{t0}\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0 - \overline{\varepsilon}_{i,j,k,n}^{xyz}\frac{\overline{k}_{i,j,k,n}^{xyz}}{\overline{k}_{i,j,k,n-1}^{xyz}} + F_{71}\overline{k}_{i,j,k,n}^{xyt} + F_{72}\overline{k}_{i,j,k-1,n}^{xyt} + F_{73} + F_{74}\overline{k}_{i-1,j,k,n}^{yzt} + F_{75}\overline{k}_{i,j,k,n}^{zxt} + F_{76}\overline{k}_{i,j-1,k,n}^{zxt}. \tag{3.26}$$

In the above equation, source term (f_k) is used from Eq. (2.14). However, it is modified by multiplying the cell averaged k at the current time step and dividing by the similarly averaged k at the previous time step. Equation (3.26) can now be written as follows:

$$\left(F_{77} + \frac{\overline{\varepsilon}_{i,j,k,n}^{xyz}}{\overline{k}_{i,j,k,n-1}^{xyz}}\right)\overline{k}_{i,j,k,n}^{xyz} - F_{78}\overline{k}_{i,j,k,n-1}^{xyz} = 2\nu_{t0}\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0 + F_{71}\overline{k}_{i,j,k,n}^{xyt} + F_{72}\overline{k}_{i,j,k-1,n}^{xyt} + F_{73}\overline{k}_{i,j}^{yz} + F_{74}\overline{k}_{i-1,j,k,n}^{yzt} + F_{75}\overline{k}_{i,j,k,n}^{zxt} + F_{76}\overline{k}_{i,j-1,k,n}^{zxt}. \tag{3.27}$$

The advantage of using this formulation is that the source term in the above equation, $2\nu_{t0}\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0$, is always positive during the transients. This precludes the possibility of negative values of k during the transient. Similar modification in ε equation results in the following discretized equation:

$$\left(F_{77}+g_{\varepsilon}C_{2\varepsilon}\frac{\overline{\varepsilon}^{xyz}_{i,j,k,n-1}}{\overline{k}^{xyz}_{i,j,k,n-1}}\right)\overline{\varepsilon}^{xyz}_{i,j,k,n}-F_{78}\overline{\varepsilon}^{xyz}_{i,j,k,n-1}=2C_{1\varepsilon}\frac{\overline{\varepsilon}^{xyz}_{i,j,k,n-1}}{\overline{k}^{xyz}_{i,j,k,n-1}}\nu_{t0}\left(\overline{s}_{mp}\overline{s}_{mp}\right)_0+F_{71}\overline{\varepsilon}^{xyt}_{i,j,k,n}+\cdots$$
$$+F_{74}\overline{\varepsilon}^{yzt}_{i-1,j,k,n}+F_{75}\overline{\varepsilon}^{zxt}_{i,j,k,n}+F_{76}\overline{\varepsilon}^{zxt}_{i,j-1,k,n} \tag{3.28}$$

Similarly, to prevent ε becoming negative during the transient, the boundary condition (Eq. (3.16)) is approximated as follows:

$$\varepsilon_{wall}=2\nu\frac{k_1}{n_1^2}\ \text{or}\ \varepsilon_{wall}=2\nu\frac{\partial\sqrt{k}}{\partial n^2} \tag{3.29}$$

where k_1 and n_1 are the turbulence kinetic energy and the distance from the wall of the interface of the first and second cells in the wall normal direction, respectively. In the present work, first of the two boundary conditions is used. For a wall parallel to the y-z plane the ε boundary condition is implemented as follows:

$$\overline{\varepsilon}^{yzt}_{i-1,j,k}=2\nu\frac{\overline{k}^{yzt}_{i,j,k}}{2a_i}\ . \tag{3.30}$$

It may be recalled (from Chapter 2) that a_i is the length of a cell (i ,j, k) in the x-direction. Here, the left surface of the (i ,j, k) cell is assumed to be on the wall. The boundary conditions for walls in the other directions can be similarly implemented.

3.4.2 Implementation of High Reynolds Number *k-ε* Model using MNIM

The momentum equation for the velocity parallel to the wall in the cells next to the walls has to be solved without the convection terms. This can be implemented by modifying the F coefficients and f term in that particular cell. For the case in which u-component of the velocity is parallel to the wall, this can be achieved by forcing u_p and

u_0 to be zero in that cell (see Eqs. (2.24-2.26)). Implementation is similar for the walls parallel to the other directions.

For the flow in the x-direction near a wall parallel to the *x*-*z* plane, Eq. (3.19) is written as follows:

$$\frac{\overline{u}^{zxt}_{i,j,k} - \overline{u}^{zxt}_{i,j-1,k}}{u_\tau} = \frac{1}{\kappa}\log\frac{2b_j\nu}{u_\tau} + 5.5 \tag{3.31}$$

Here, the left surface of the (*i* ,*j*, *k*) cell is assumed to be on the wall. Similar equations can be written for walls parallel to any other direction. The above equation is a non-linear equation to be solved for u_τ. The Newton-Raphson iteration scheme is used to solve this equation. Following function is defined for the Newton iterations:

$$G(u_\tau) = \frac{\overline{u}^{zxt}_{i,j,k} - \overline{u}^{zxt}_{i,j-1,k}}{u_\tau} - \frac{1}{\kappa}\log\frac{2b_j\nu}{u_\tau} + 5.5 \tag{3.32}$$

and the iterations are then implemented as follows:

$$u_\tau^{m+1} = u_\tau^m - \frac{G\left(u_\tau^m\right)}{G'\left(u_\tau^m\right)} \tag{3.33}$$

where, *m* is the iteration number and $G'(u_\tau)$ is the derivative of the function $G(u_\tau)$ with respect to u_τ. The iterations are performed till $G(u_\tau)$ is reasonably small (less than a pre-specified limit). Though, subscript $(i,\ j,\ k)$ are not used for u_τ, it should be noted that it needs to be evaluated for each cell next to the wall. Once u_τ is evaluated, the implementation of the boundary conditions for the *k* and *ε* equations is straightforward.

3.6 Results for Low Reynolds Number Model

The MNIM implementation of the k-ε model is tested using the fully developed channel flow. Although the problem is one-dimensional, it is a good test case because the low Reynolds number k-ε models are generally calibrated using this problem so as to match the channel flow characteristics. Therefore, successful implementation of the k-ε model using MNIM should be able to produce good match with DNS results (or with experimental) data. The scheme is implemented on 2D version of MNIM with periodic boundary conditions for the velocity, k and ε in the streamwise direction. A constant pressure gradient is applied at the inlet of the channel and a constant pressure is maintained at the outlet. The results are presented here for two Reynolds numbers; Re_τ = 180 and Re_τ = 550. The Re_τ is friction Reynolds number and is defined as follows for the channel flow:

$$\mathrm{Re}_\tau = \frac{\nu u_\tau}{\delta}. \tag{3.34}$$

The results are compared with the DNS results available at the website (http://torroja.dmt.upm.es/ftp/channels/data/) mentioned in the work of del Alamo (2003).

3.6.1 Channel Flow with Re_τ = 180

The velocity, turbulent kinetic energy and dissipation rate plots are given in Fig. 3.2, Fig. 3.3 and Fig. 3.4, respectively. The DNS results are also plotted for comparison. Two different non-uniform grids are used for the solution of this problem. The first grid has 6 cells in the wall normal direction and the successive ratio of the length of the cells

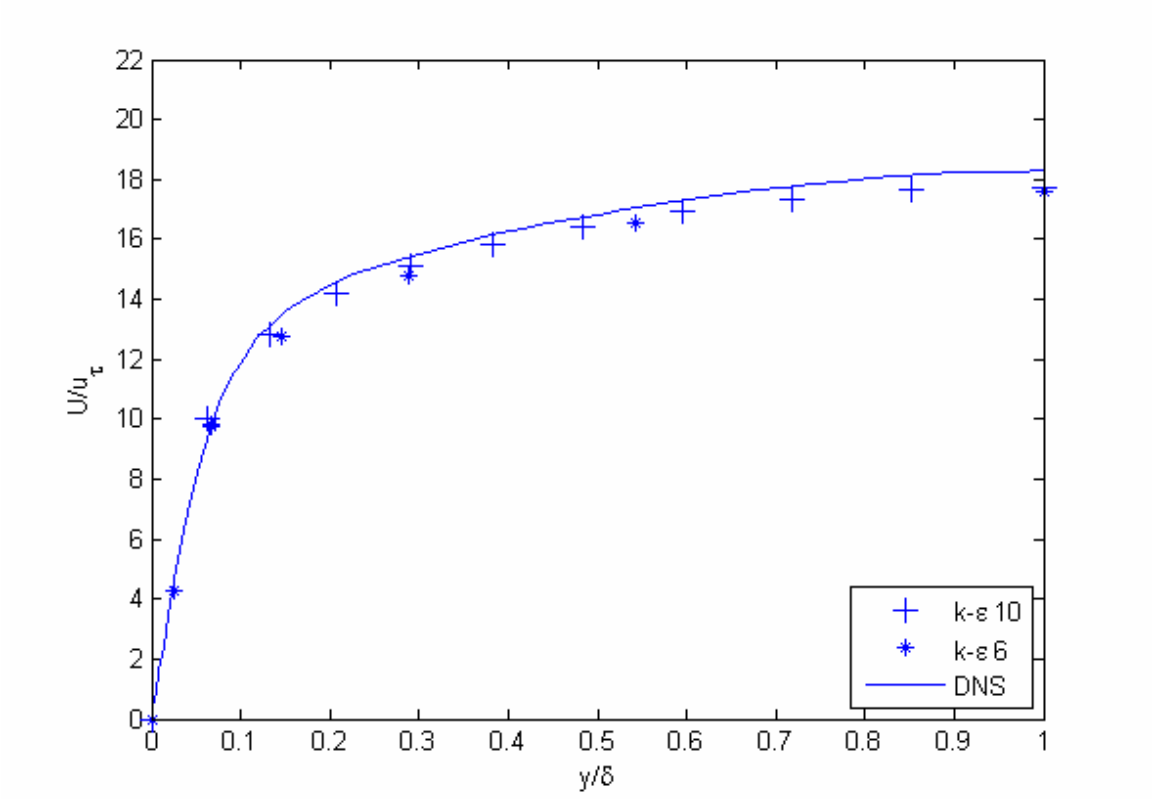

Fig. 3.2 Comparison of axial velocity distribution obtained using the low Reynolds number k-ε model and the DNS results (for Re_τ =180). The k-ε model results are reported for two different grids (6 cells in the wall normal direction and 10 cells in the wall normal direction). There is good agreement with the DNS results

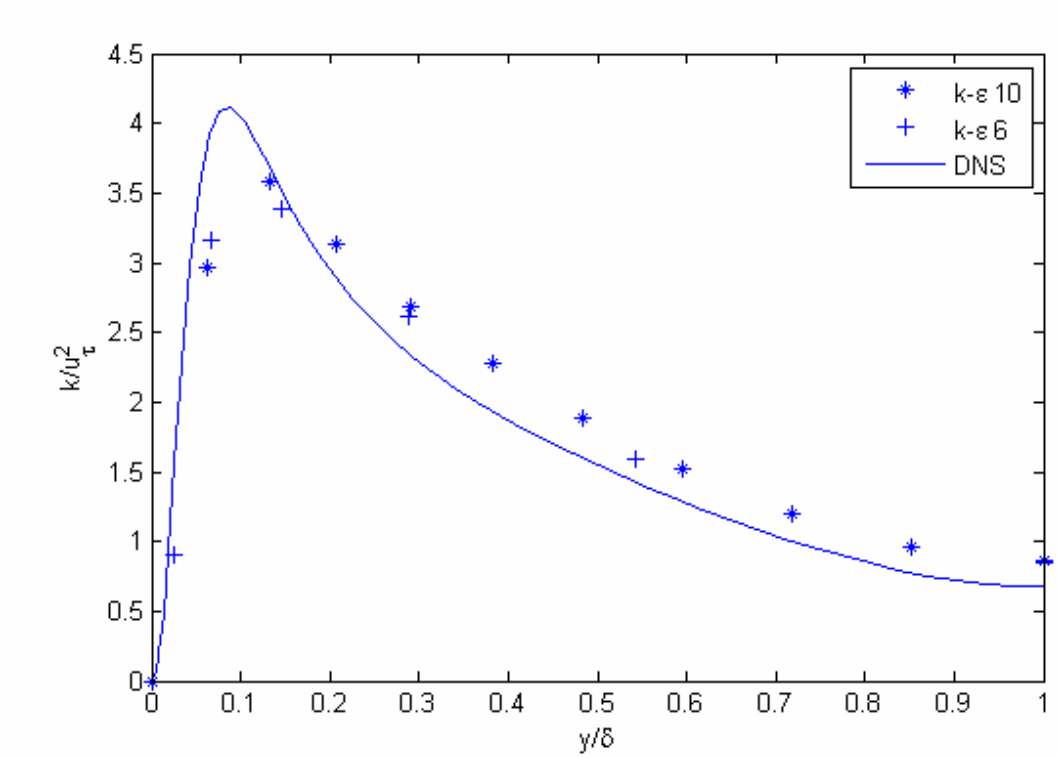

Fig. 3.3 Comparison of turbulent kinetic energy obtained using the low Reynolds number k-ε model and the DNS results (for Re_τ =180). The k-ε model results are reported for two different grids (6 cells in the wall normal direction and 10 cells in the wall normal direction). There is good agreement with the DNS results.

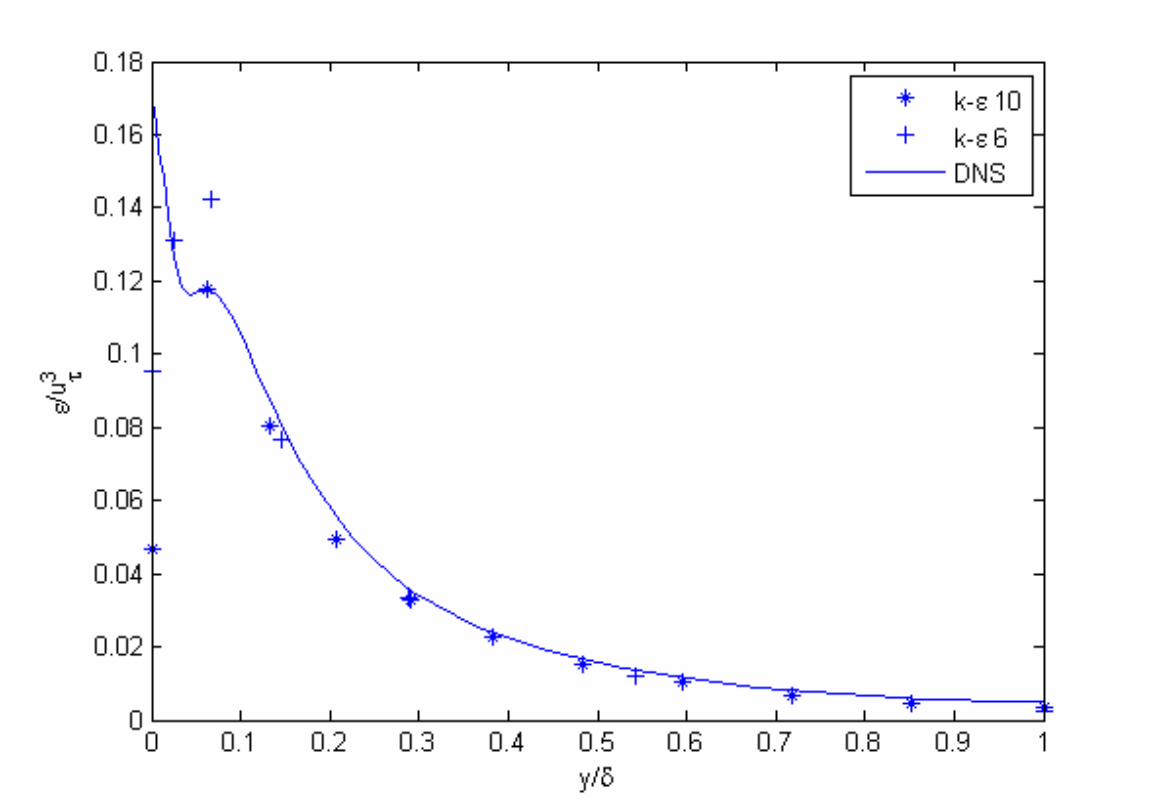

Fig. 3.4 Comparison of dissipation of turbulent kinetic energy obtained using the low Reynolds number k-ε model and the DNS results (for Re_τ =180). The k-ε model results are reported for two different grids (6 cells in the wall normal direction and 10 cells in the wall normal direction). There is good agreement with the DNS results.

in the wall normal direction is 1.7. Second grid has 10 cells in the wall normal direction and abovementioned successive ratio is 1.1.

3.6.2 Channel Flow with $Re_\tau = 550$

The same number of cells are used for the simulation of the flow with $Re_\tau = 550$ as that for the $Re_\tau = 180$. However, because of the thinner boundary layers the grids usedare more non-uniform. The successive grids with 10 cells in the wall normal direction have a ratio of 1.2 and that with 6 cells have a ratio of 1.8. Figures 3.5, 3.6 and 3.7 show the velocity, k and ε, respectively, for this Reynolds number case.

Overall, the results compare very well with the DNS results for all the three quantities compared and for both Reynolds numbers. The ε values near the wall show a peak, which is different from the behavior observed in the DNS results. However, the AKN model is known to show this discrepancy in the ε values (Park 1994).

Since low Reynolds number model are constructed so as to match the behavior of the flow in the viscous layer near the wall, it is necessary to have grid points inside the thin viscous layer for the low Reynolds number model. This also explains the fact that very coarse grid away from the wall also show good agreement with DNS results as long as the near wall region is resolved reasonably well. Therefore, near wall grid resolution required, in general, is independent of the numerical scheme used to implement the model.

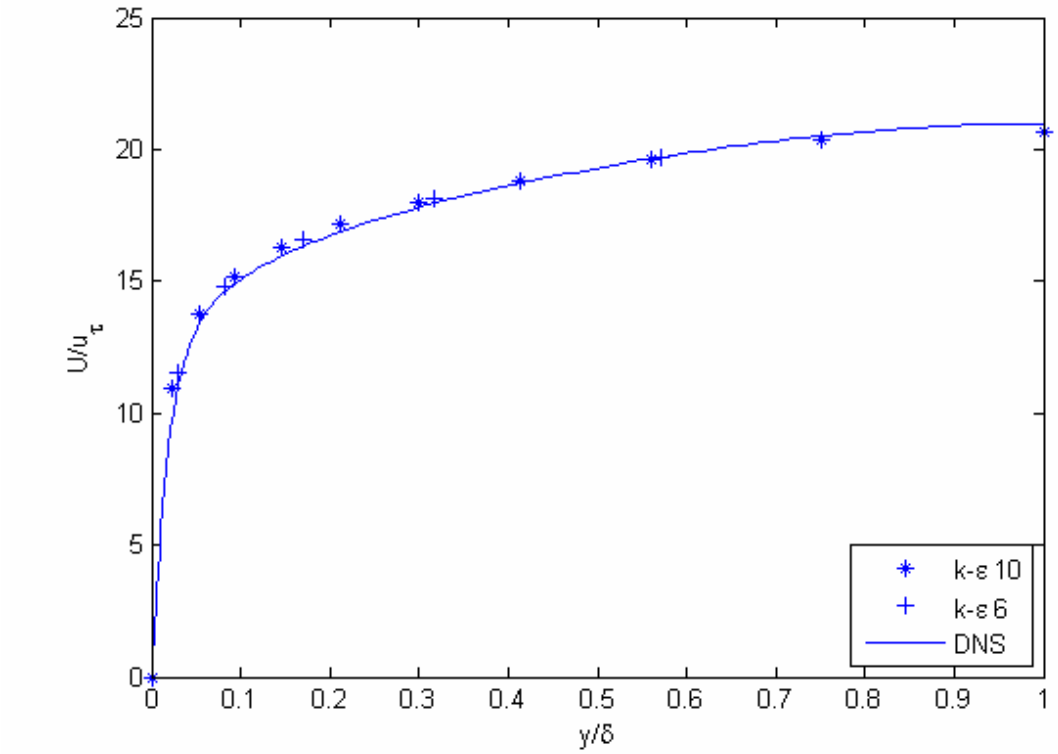

Fig. 3.5 Comparison of axial velocity distribution obtained using the low Reynolds number k-ε model and the DNS results (for $Re_\tau = 550$). The k-ε model results are reported for two different grids (6 cells in the wall normal direction and 10 cells in the wall normal direction). There is good agreement with the DNS results.

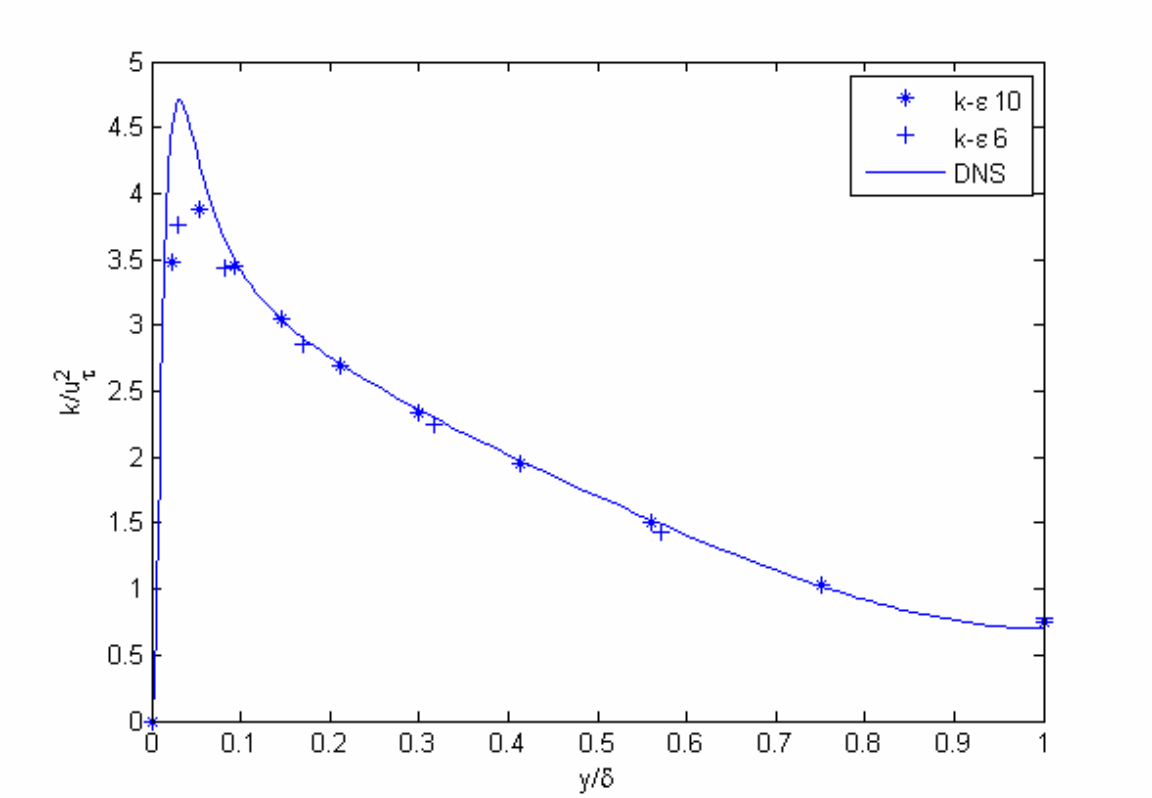

Fig. 3.6 Comparison of turbulent kinetic energy obtained using the low Reynolds number k-ε model and the DNS results (for $Re_\tau = 550$). The k-ε model results are reported for two different grids (6 cells in the wall normal direction and 10 cells in the wall normal direction). There is good agreement with the DNS results.

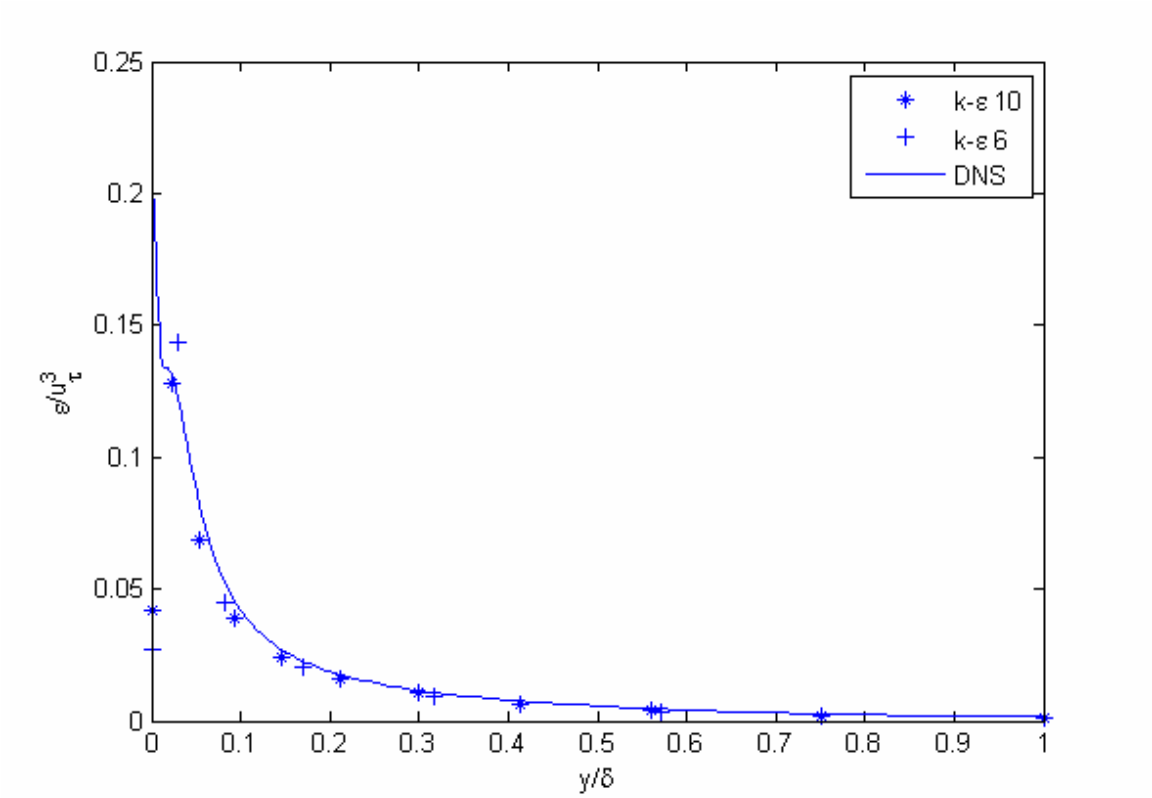

Fig. 3.7 Comparison of dissipation of turbulent kinetic energy obtained using the low Reynolds number k-ε model and the DNS results (for $Re_\tau = 550$). The k-ε model results are reported for two different grids (6 cells in the wall normal direction and 10 cells in the wall normal direction). There is good agreement with the DNS results.

3.7 Results for High Reynolds Number Model

3.7.1 Channel Flow with $Re_\tau = 2000$

The channel flow with relatively high Reynolds number i.e. $Re_\tau = 2000$ is used for testing the high Reynolds number model. Since, u_τ is known *a priori* for the channel flow with a given Re_τ, therefore solving the non-linear equation for u_τ (Eq. (3.28)) is not necessary. However, to test the implementation, the flow equations are solved both with Eq. (3.28) and without it. For the case where the u_τ is assumed to be known, the equations are solved only in the region which lies between $y^+ = 50$ ($y/\delta = 0.025$) and the center of the channel. The Dirichlet boundary conditions for the k and ε, which are given by Eq. (3.20) and Eq.(3.21) respectively, are applied on the surface of the first cell (at $y/\delta = 0.025$). The values of the k and ε at this surface can be computed using the above mentioned equations because it lies in the log-law region. Similarly, Dirichlet boundary condition is applied for the momentum equation at the same surface with velocity given by Eq. (3.19). Figures 3.8, 3.9 and 3.10 compare the result for velocity, k and ε, respectively, obtained using the above scheme with the DNS results. The DNS results used here are stored at the website (http://torroja.dmt.upm.es/ftp/channels/data/) cited by Hoyas (2006).

Simulations have been carried out with 10, 20 and 30 cells in the wall normal direction. All the grids used are uniform in this case. The results match very well with the DNS results, especially for the two finer grids. The k values show some discrepancy from the DNS results; however, this discrepancy is due to the shortcoming in the model used and not from the numerical scheme. The reason for this discrepancy is that k is not constant in the log-layer as assumed in the model.

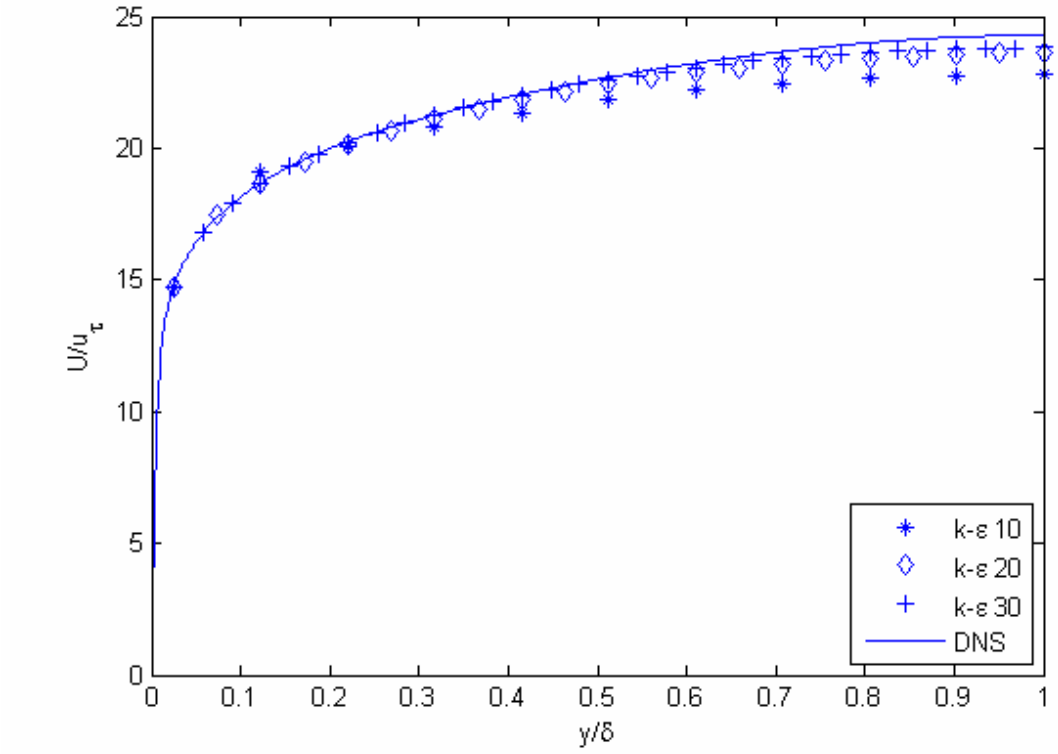

Fig. 3.8 Comparison of axial velocity distribution obtained using the high Reynolds number k-ε model and the DNS results (for $Re_\tau = 2000$). The results are obtained without the calculation of the u_τ.The k-ε model results are reported for three different grids (10 , 20 and 30 cells in the wall normal direction, respectively). There is good agreement with the DNS results.

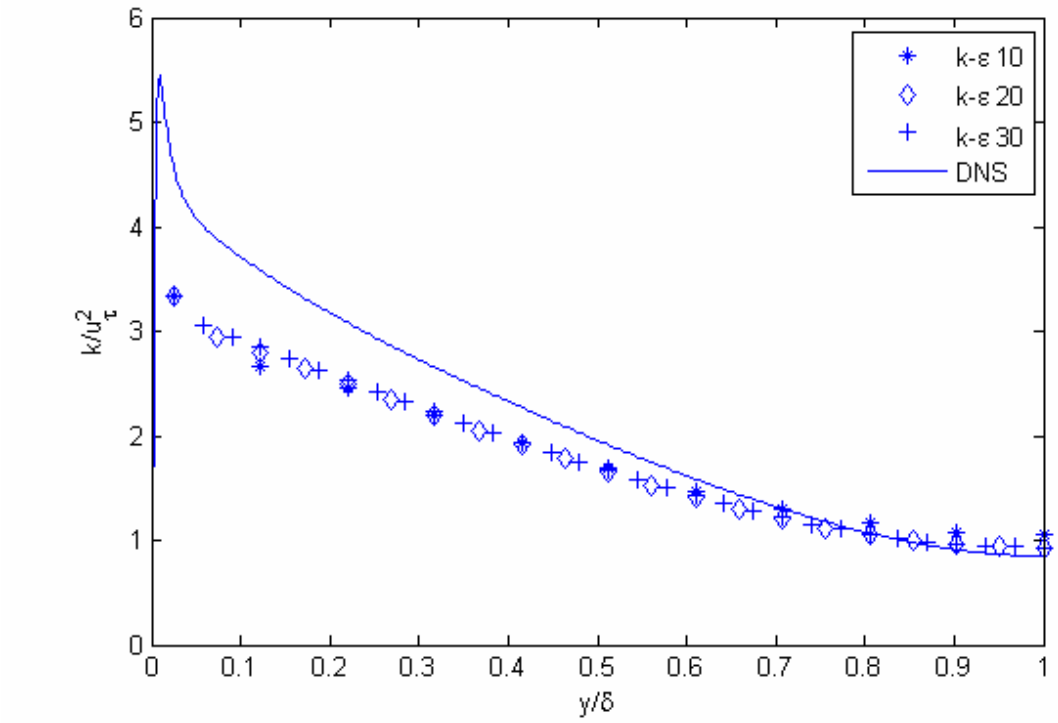

Fig. 3.9 Comparison of turbulent kinetic energy. obtained using the high Reynolds number k-ε model and the DNS results (for $Re_\tau = 2000$). The results are obtained without the calculation of the u_τ. The k-ε model results are reported for three different grids (10 , 20 and 30 cells in the wall normal direction, respectively).

.

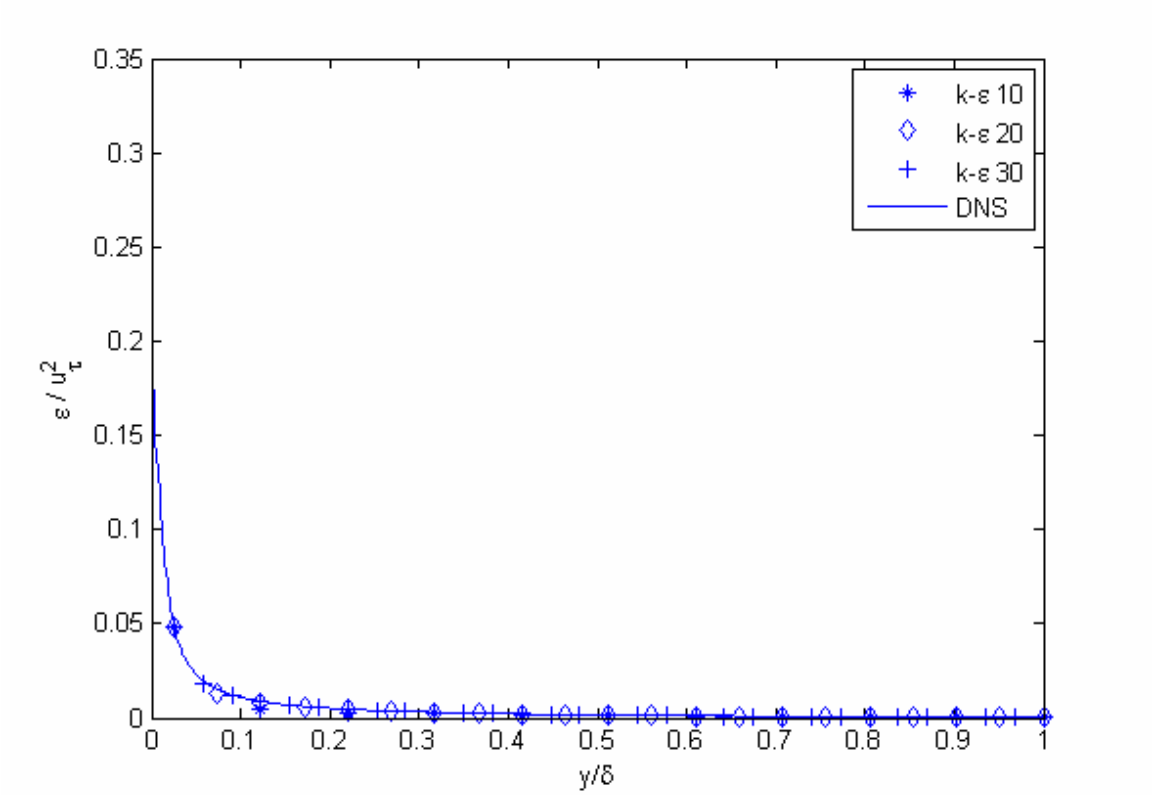

Fig. 3.10. Comparison of dissipation of turbulent kinetic energy obtained using the high Reynolds number k-ε model and the DNS results (for $Re_\tau = 2000$). The results are obtained without the calculation of the u_τ.The k-ε model results are reported for three different grids (10, 20 and 30 cells in the wall normal direction, respectively). There is good agreement with the DNS results.

In the second case, u_τ is computed using Eq. (3.28). The equations are solved over the entire channel, and the boundary conditions are applied as discussed previously for the high Reynolds number model. First two grids used for this case are uniform with 10 and 20 cells in the wall normal direction. Third grid used is a non-uniform grid with only 10 cells in the wall normal direction. The ratio between the dimensions of the successive cells in the wall normal direction is 1.1. Non-uniform grids are used making sure that the interface in Fig. 2.1 lies in the log-law region. The results obtained are similar to those that have been obtained without calculating the u_τ. The results are shown in the Figs. 3.11, 3.12, and 3.13 for velocity, k and ε, respectively. The non-uniform 10 cell grid results are as good as 20 cell grid. Both these results match very well with the DNS results except for some discrepancy in k values. The reason for discrepancy in k values has been discussed previously. The similarity in results obtained with the calculation of u_τ and those without that calculation show that the correct values of u_τ are being obtained by solving the non-linear equation.

3.7.2 Lid Driven Cavity with Re = 20,000

The high Reynolds number model is used to solve the lid driven cavity problem. The schematic and boundary conditions for the velocities are given in the Fig. 3.14. A non-uniform 30 x 30 grid is used for the simulation. The plot of the u-velocity at the vertical mid-plane and the v-velocity at the horizontal mid-plane are shown in Figs. 3.15(a) and 3.16(a), respectively. Figures 3.15(b) and 3.16(b) show the plots of DNS results for similar 3D cubic cavity for Re = 12000. These latter figures are taken from Boufannais (2007) who carried out LES simulations of the lid driven cavity problem. The

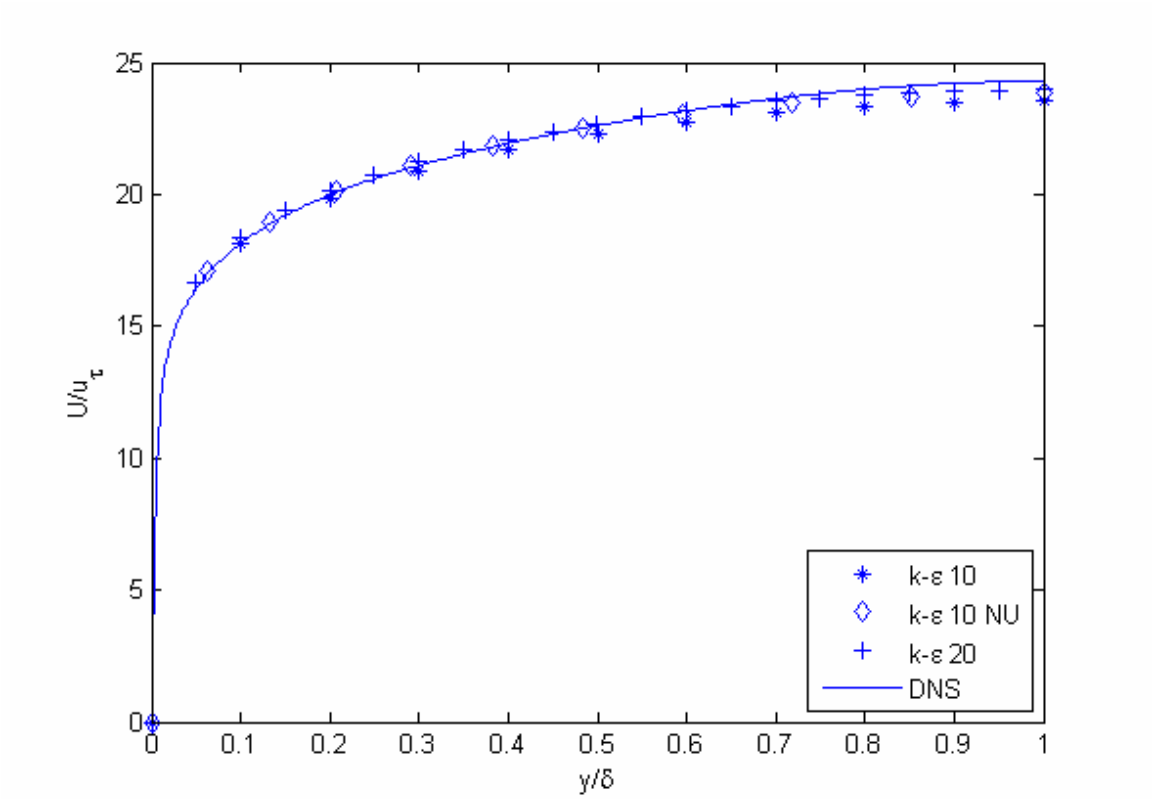

Fig. 3.11 Comparison of axial velocity distribution obtained using the high Reynolds number k-ε model and the DNS results (for $Re_\tau = 2000$). The results are obtained withthe calculation of the u_τ.The k-ε model results are reported for three different grids (10, 20 and 30 cells in the wall normal direction, respectively). There is good agreement with the DNS results.

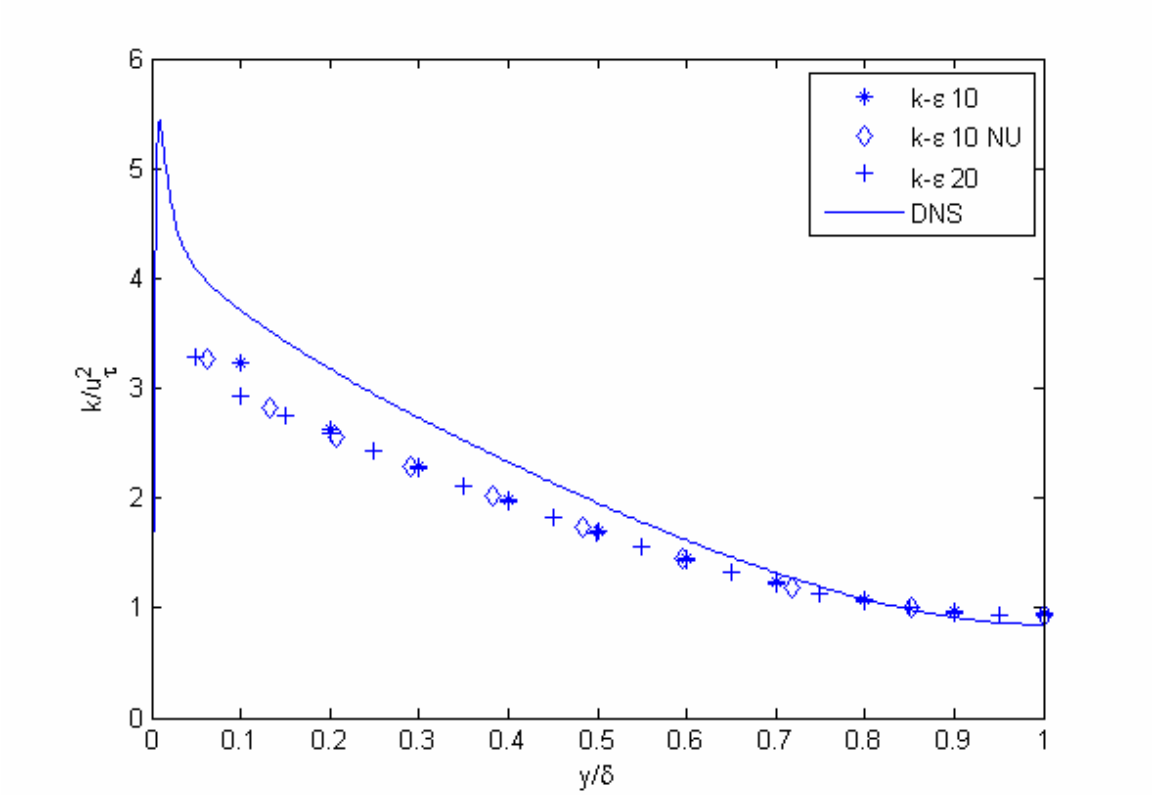

Fig. 3.12 Comparison of turbulent kinetic energy obtained using the high Reynolds number k-ε model and the DNS results (for $Re_\tau = 2000$). The results are obtained with the calculation of the u_τ.The k-ε model results are reported for three different grids (10, non – uniform 10 and 30 cells in the wall normal direction, respectively). There is reasonably good agreement with the DNS results.

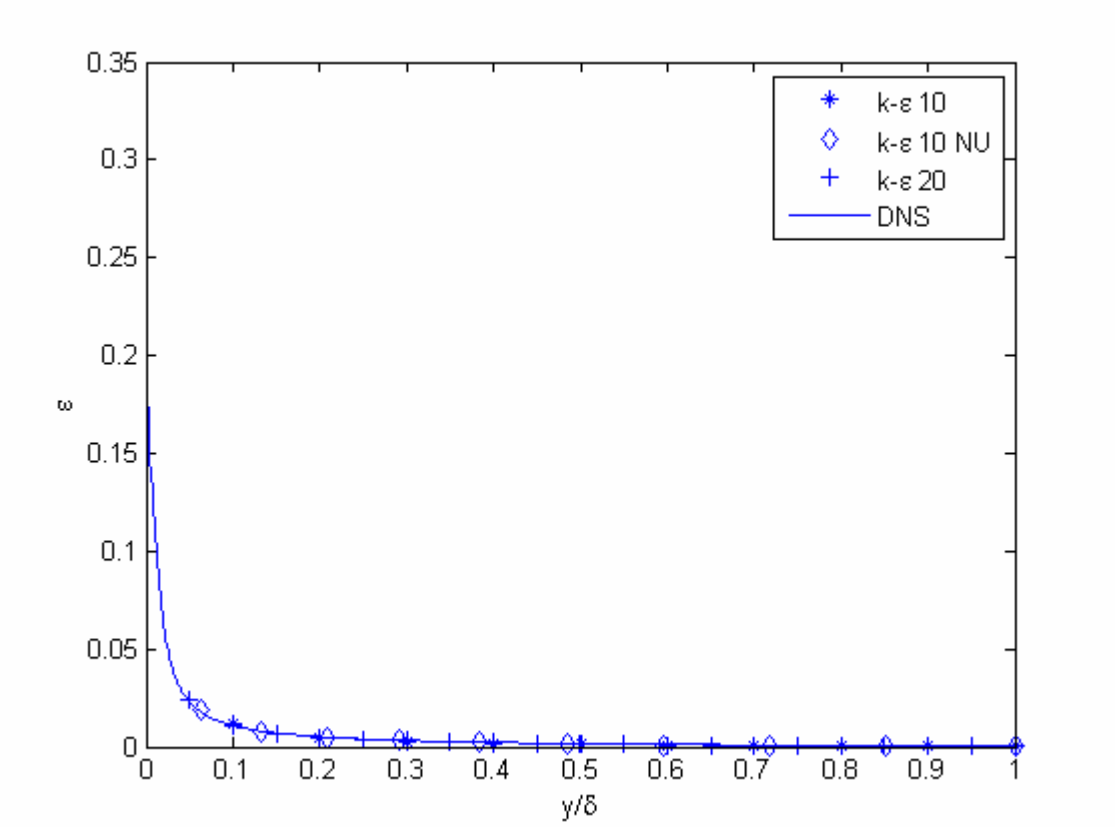

Fig. 3.13 Comparison of dissipation of kinetic energy obtained using the high Reynolds number k-ε model and the DNS results (for $Re_\tau = 2000$). The results are obtained with the calculation of the u_τ.The k-ε model results are reported for three different grids (10, 20 and 30 cells in the wall normal direction, respectively). There is good agreement with the DNS results.

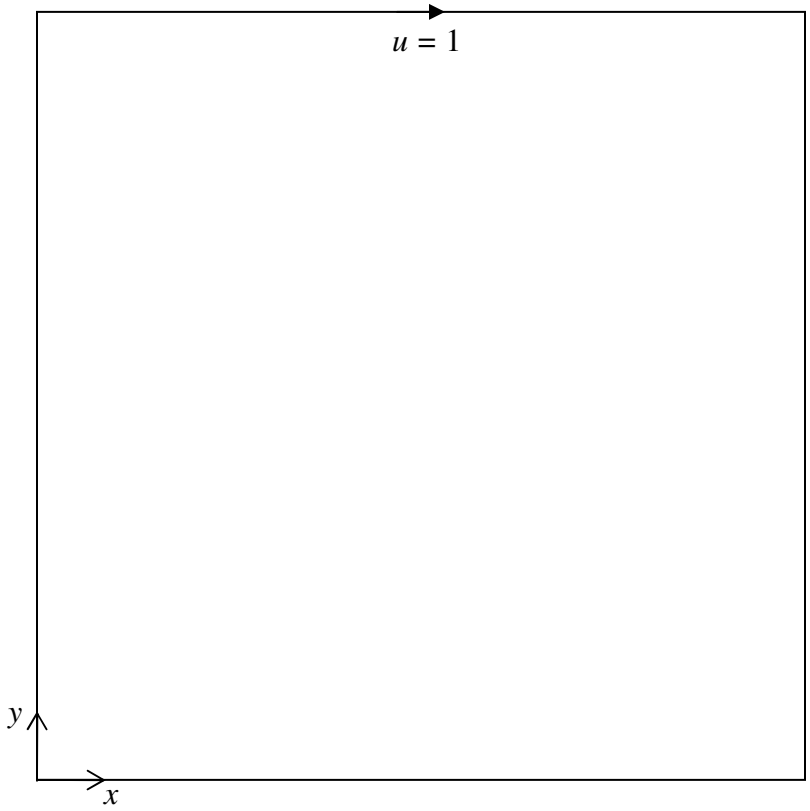

Fig. 3.14 Schematic of a lid driven cavity of unit dimensions.

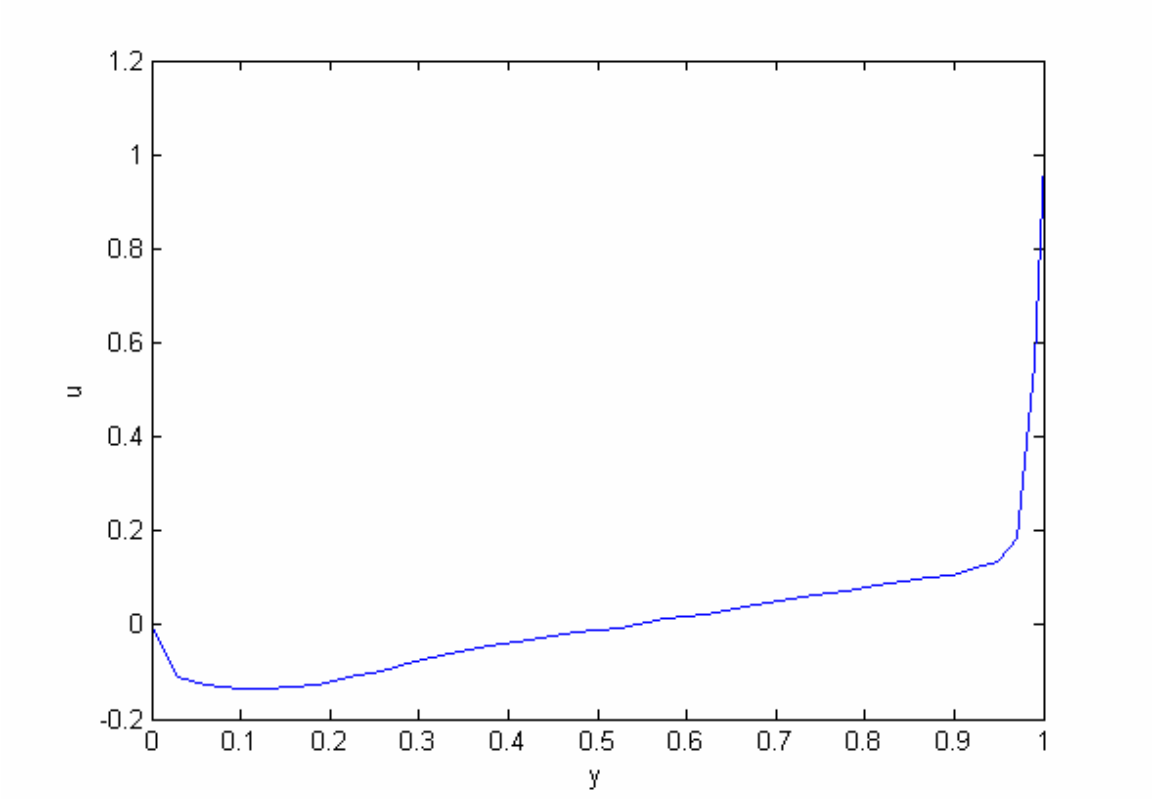

Fig. 3.15 (a) Plot of *u*-velocity for lid driven cavity at vertical mid-plane obtained using the high Reynolds number *k*-*ε* model.

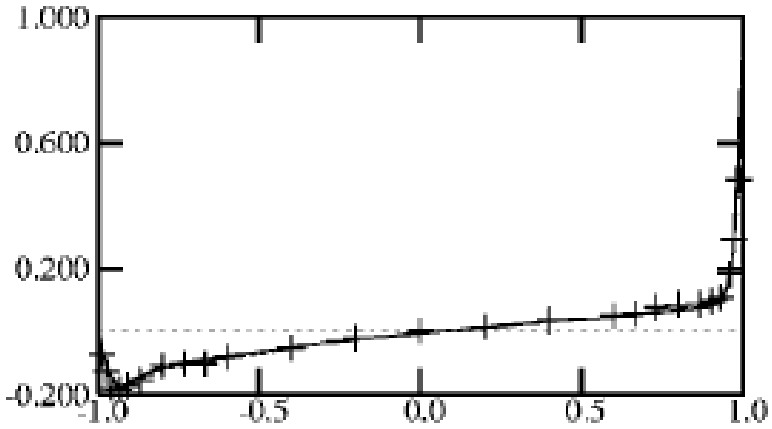

Fig. 3.15 (b) Plot of *u*-velocity for lid driven cavity at the vertical mid-plane. Results were obtained using the DNS approach (solid line). (The figure is taken from Bouffanais (2007))

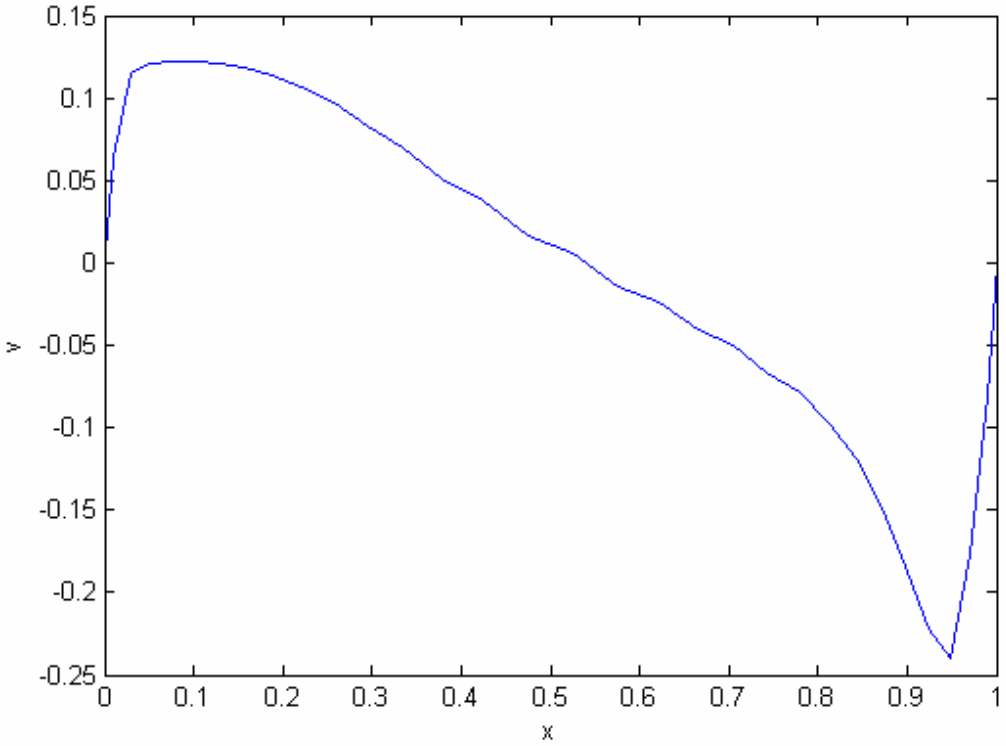

Fig. 3.16 (a) Plot of *v*-velocity for lid driven cavity at horizontal mid-plane obtained using the high Reynolds number *k*-ε model.

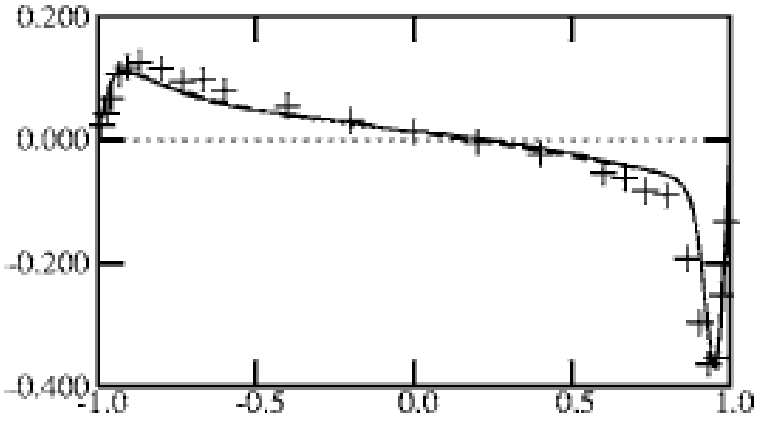

Fig. 3.16 (b) Plot of *v*-velocity for the lid driven cavity at the horizontal mid-plane. Results were obtained using the DNS approach (solid line). (The figure is taken from Bouffanais (2007))

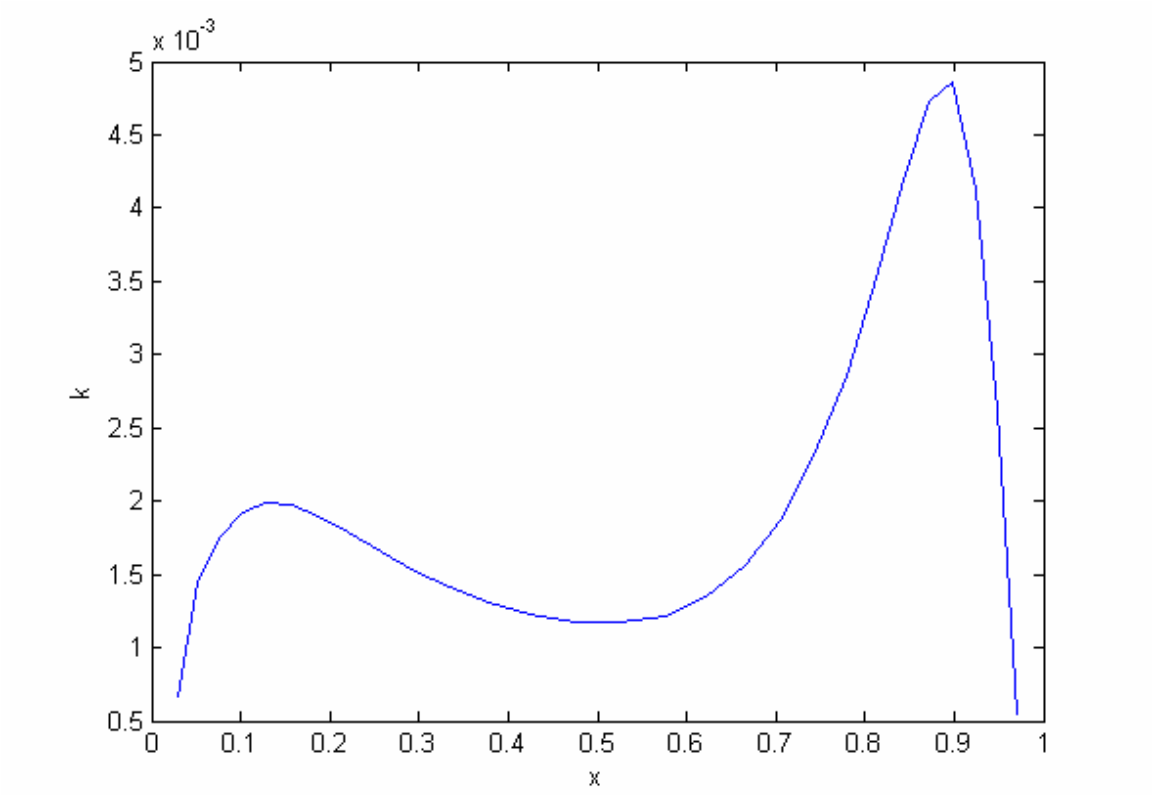

Fig. 3.17 Plot of k for the lid driven cavity at the vertical mid-plane obtained using the high Reynolds number k-ε model.

.

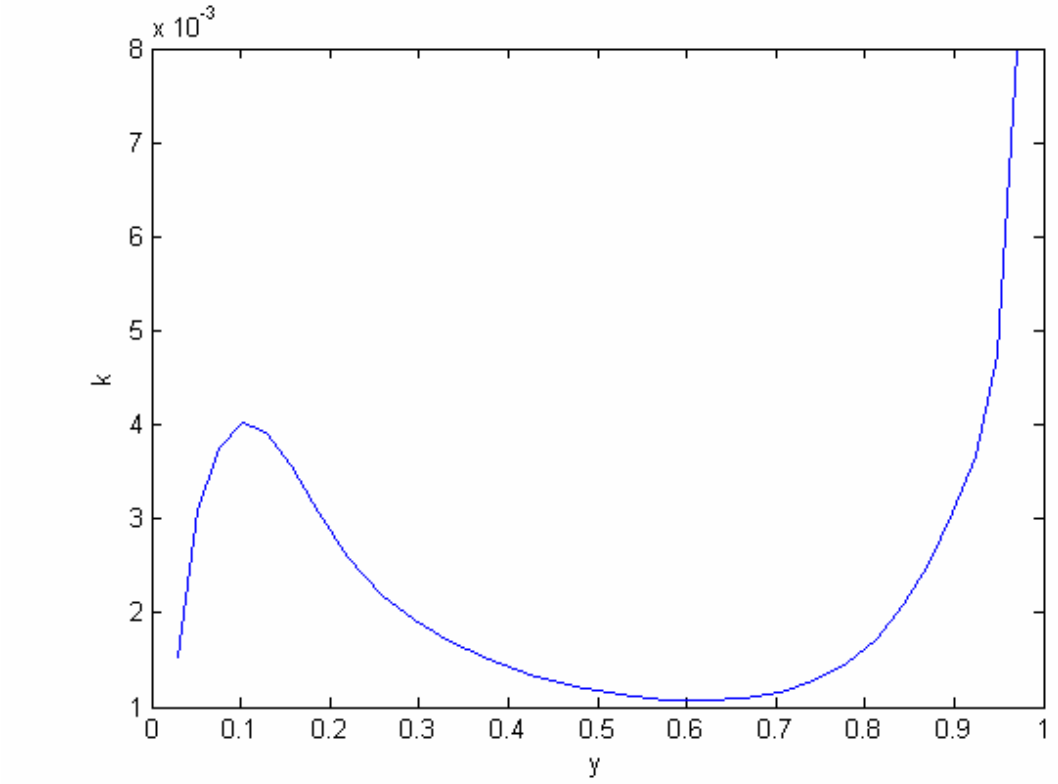

Fig. 3.18 Plot of k for the lid driven cavity at the horizontal mid-plane obtained using the high Reynolds number k-ε model.

.

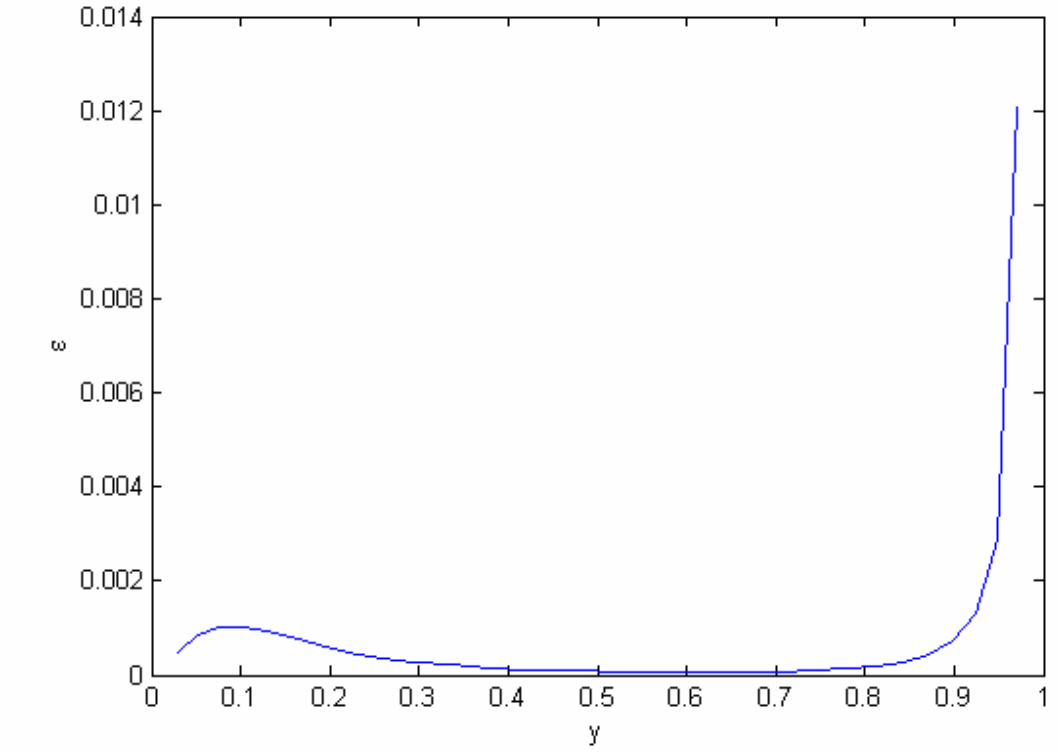

Fig. 3.19 Plot of ε for the lid driven cavity at the vertical mid-plane obtained using the high Reynolds number k-ε model.

.

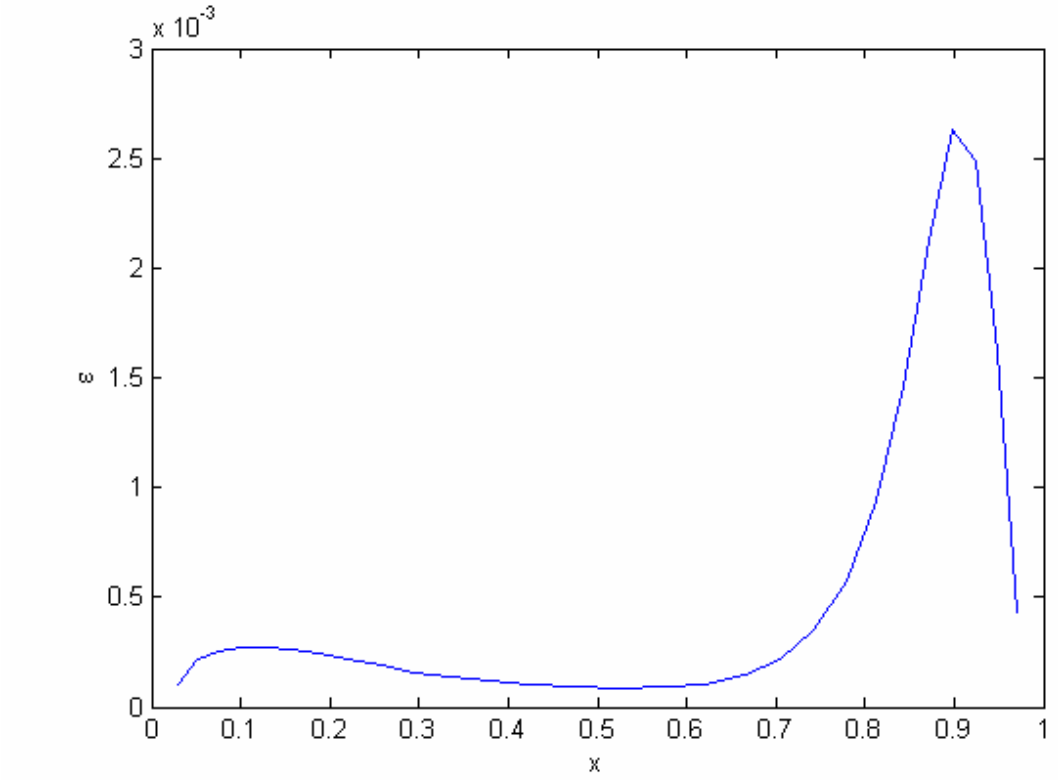

Fig. 3.20 Plot of ε for the lid driven cavity at the horizontal mid-plane obtained using the high Reynolds number k-ε model.

DNS results shown in those figures are from Leriche (2000). The DNS results at Re = 12000 are shown because the results for higher Re are not available. The k-ε model is not used to simulate the flow for Re = 12,000 because at this Re the flow in most of the domain is laminar (Leriche 2000). In addition to the plots of the velocity, the k at the vertical mid-plane is plotted in Fig. 3.17 and that at the vertical mid-plane is shown in Fig. 3.18. The corresponding ε plots are shown in Figs. 3.19 and 3.20, respectively. The DNS results for the k and ε values are not available. However, the plots for these values are qualitatively similar to the plots given by the Nonino (1988) (not shown here). The direct comparison is not possible because of very coarse grid used in the abovementioned reference.

It can be seen from the plots of velocity, although the values of the velocity are significantly different, DNS and k-ε model have qualitatively similar results. The significant differences in these results are, however, expected because of various assumptions made in the k-ε model. Also, the spatial variation of k matches qualitatively with the results reported by Nonino (1988) (not shown here). Direct comparison with those results is not possible because of very coarse grid used by Nonino (1988).

3.8 Summary of the Implementation of k-ε Model

The implementation of the k-ε model using the MNIM formulation is discussed in this chapter. Two models, one for low Reynolds number and one for high Reynolds number, are implemented. However, different variations of these models can be implemented in a similar manner as underlying features of these models are quite similar. Moreover, the treatment of regions away from the walls in the k-ε models is largely the

same (except for some differences in the model constants), irrespective of the different models used.

The application of the k-ε model to relatively simple problems shows that results are in good agreement with those reported in literature and similar to what one would expect from the k-ε model implementation with any other numerical scheme. The results are compared with the DNS results from various sources in the literature. The difference between the DNS results and current implementation arises mainly from the assumptions made in the k-ε model and hence the limitations of the k-ε model rather than the choice of the numerical scheme in the present work. The above observation is based on the fact that, in general, the discrepancies seen in the present work are same as those reported for the simulations carried out using other numerical schemes.

It is seen that very coarse grids can be used away from the walls for the present simulation. This is especially true for low Reynolds number model. In the high Reynolds number case, the near wall grid resolution is restrained by the fact that the interface between the first and second cell must lie on the log-law region. Similarly, in the low Reynolds number model, the resolution of the near wall viscous layer is necessary. Therefore, near wall grid refinement is more dependent on the k-ε model used, than on the choice of the numerical scheme used for the simulation. However, away from the wall relatively few cells are needed.

Chapter 4

Parallel Modified Nodal Integral Method

It is natural to use MNIM to simulate turbulent flows because of its high efficiency. However, it is necessary to first develop a parallel version of the MNIM for the simulation of the turbulent flows which are computationally highly intensive.

Various parallelization strategies have been developed for unstructured and structured grids for different numerical methods. For example, several computational strategies have been developed for parallelization of pseudo spectral methods which are, because of their high numerical accuracy, particularly suitable for direct numerical simulations, although for limited geometrical configurations. These have been implemented on several different types of machines. For example, Pelz (1991) implemented parallel spectral methods for the Navier-Stokes equations on a 1024-node hypercube computer. Jackson (1991) and Chen (1992) implemented it on an Intel iPSC/860 hypercube machine and CM-2 machine, respectively. Basu (1994) implemented a pseudospectral scheme on a three-processor multicomputer. Numerous parallel implementations of other numerical schemes – finite difference, finite element, control volume – have also been reported. Levit (1988) used finite difference based parallel solvers for their flow simulations. Naik (1993) also developed parallel finite difference fluid solvers. Mixed spectral and finite difference schemes were implemented by Prestin (1995) and also by Garg (1995) on parallel computers. Wu (1997) presented an

adaptive parallel multigrid method for the incompressible Navier–Stokes equations. Passoni (2001) developed a parallelization strategy for shear flow simulations. Finite element based parallel solver for fluid flow was implemented by Johan (1992). Wasfy (1998) developed a parallel finite element scheme for incompressible Navier-Stokes equation. Compressible flow simulations were carried out by Mittal (1995) using parallel finite element method. Recently, Liu (2003) developed a parallel Galerkin FEM for flow simulations.

In this chapter, development and implementation of a parallel MNIM for the Navier-Stokes equations using the domain decomposition paradigm for a structured grid is discussed. The dependencies of the variables in the subdomains, which result from the decomposition of the original computational domain, on the variables in the neighboring subdomains are identified and communication procedure is accordingly implemented. Message Passing Interface (MPI) (Gropp 1999), which is a library specification for message-passing, is used for domain decomposition as well as for communications between processors. The parallel version of the MNIM developed here is tested by applying it to several test problems. Speedup and efficiency for different number of processors are presented for one test problem.

4.1 Dependencies in the Equations of MNIM

The dependencies of various variables on each other and the template dictate the scheme that can be used for proper communication of the variables between the processors in the parallel version of the MNIM. The templates for the various variables

are discussed in the previous chapter. It should be noted that for this purpose the template of the momentum equations—with the exception of the *f* term in Eq. (2.14)—can be considered to be the same as the template for the convection-diffusion equation. Therefore, dependencies of only the convection-diffusion equation and pressure equation need to be considered.

Figure 4.1 shows the dependencies in the equation for the variable $\overline{C}^{zxt}_{i,j,k,n}$ and $\overline{C}^{xyz}_{i,j,k,n}$. These dependencies are obvious from Eqs. (2.12) and (2.14). The dependencies for $\overline{C}^{yzt}_{i,j,k,n}$ and $\overline{C}^{xyt}_{i,j,k,n}$ in the *x* and *z* directions are similar to those for $\overline{C}^{zxt}_{i,j,k,n}$. It should also be noted that the *f* term in the momentum equations, unlike its counterpart in the convection-diffusion equation, has its own dependencies. For example, $\overline{u}^{xyz}_{i,j,k,n}$ depends on $\overline{p}^{yzt}_{i-1,j,k,n}$ and $\overline{p}^{yzt}_{i,j,k,n}$ through the *f* term. Similar dependencies in the *y* and *z*- momentum equation are present in the respective directions. Moreover, it also depends on the velocity variables. However, these dependencies are the same as shown in Fig. 4.1.

Figure 4.2 shows the template for $\overline{p}^{zxt}_{i,j,k,n}$ (Eq. (2.31)). Similar dependencies for Eqs. (2.28) and (2.29) and the corresponding variables can also be obtained. It should be noted that the f_1 term in Eq. (2.31), which can be considered to be located at the center of a cell, is dependent on the surface averaged velocities of the surfaces surrounding that cell.

In addition to the templates discussed so far, the *F* coefficients in the velocity equations depend on the previous time step cell averaged velocities, i.e. on $\overline{u}_{pi,j,k,n}$, $\overline{u}_{pi+1,j,k,n}$, $\overline{v}_{pi,j,k,n}$, $\overline{v}_{pi,j+1,k,n}$, $\overline{w}_{pi,j,k,n}$ and $\overline{w}_{pi,j,k+1,n}$. The *F* coefficients in the velocity as

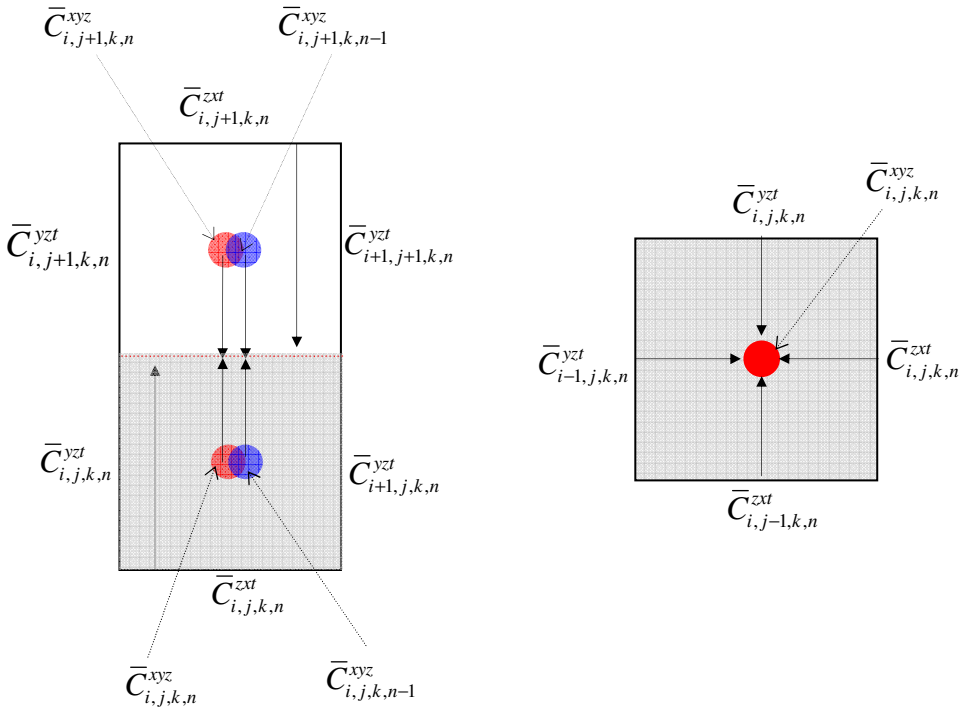

Fig 4.1. The dependencies of the computation of discrete variables $\bar{C}^{zxt}_{i,j,k,n}$ and $\bar{C}^{xyz}_{i,j,k,n}$ on the neighboring variables. The shaded cell in the figure is cell (*i, j, k*) and red (dashed) line represents variable $\bar{C}^{zxt}_{i,j,k,n}$. Only *x*-*y* plane cutting through the vertical (*z*-direction) center of the cells has been shown for the sake of clarity. Variables $\bar{C}^{xyt}_{i,j,k-1,n}$ and $\bar{C}^{xyt}_{i,j,k,n}$ (not shown here) are located at the bottom and top surfaces of the node (*i*, *j*, *k*), respectively. (a) Dependencies for $\bar{C}^{zxt}_{i,j,k,n}$; (b) Dependencies for $\bar{C}^{xyz}_{i,j,k,n}$. In addition to variables shown here dependence is also on variables $\bar{C}^{xyt}_{i,j,k-1,n}$, $\bar{C}^{xyt}_{i,j,k,n}$ and $\bar{C}^{xyz}_{i,j,k,n-1}$.

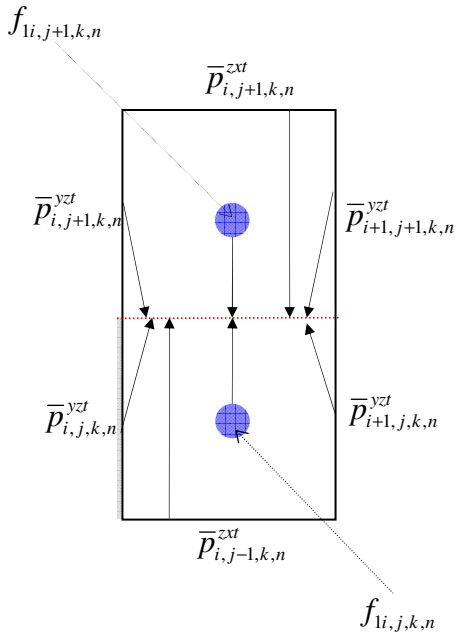

Fig 4.2. The dependencies for the computation of discrete variable $\overline{p}^{zxt}_{i,j,k,n}$ on the neighboring variables. The shaded cell in the figure is cell (*i, j, k*) and red (dashed) line represents variable $\overline{p}^{zxt}_{i,j,k,n}$. Only *x*-*y* plane cutting through the vertical (*z*-direction) center of the cells has been shown for the sake of clarity. Variables $\overline{p}^{xyt}_{i,j,k-1,n}$ and $\overline{p}^{xyt}_{i,j,k,n}$ (not shown here) are located at the bottom and top surfaces of the node (*i*, *j*, *k*), respectively. In addition to variables shown here dependence is also on variables $\overline{p}^{xyt}_{i,j,k-1,n}$, $\overline{p}^{xyt}_{i,j,k,n}$, $\overline{p}^{xyt}_{i,j+1,k-1,n}$ and $\overline{p}^{xyt}_{i,j+1,k,n}$.

well as the pressure equations also depend on the cell dimensions of the neighboring nodes.

4.2 Parallelization of MNIM

4.2.1 Domain Decomposition

Domain decomposition paradigm is a natural way of parallelization for a system where computation of a variable in a cell depends on the variables in neighboring cells. Since, aforementioned system of discrete equations when solved using iterative schemes (e.g. Gauss-Seidel, SOR, etc) is such a system; the domain decomposition is used for parallelization of MNIM.

The domain decomposition scheme is shown in Fig. 4.3. The computational domain is divided into several subdomains (one for each processor). Each processor stores and computes variables in its subdomain, exchanging information with neighboring processors when necessary.

4.2.2 Communication between Processors

The template dictates which discrete variables must be communicated to neighboring processors.. The variables thus communicated are stored in the ghost surfaces or cells in the receiving processor. Figure 4.4 shows two neighboring

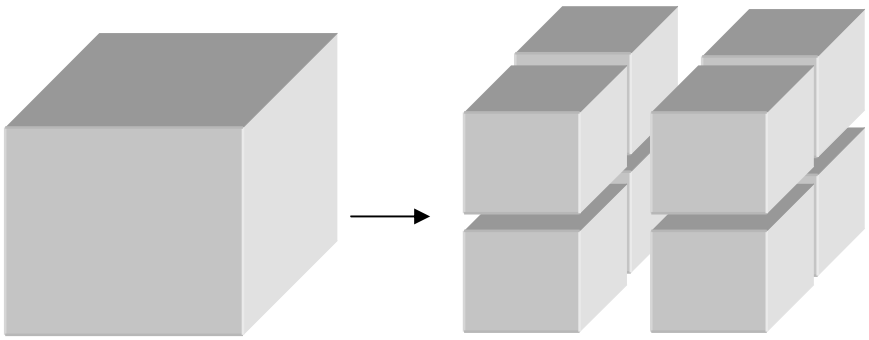

Fig. 4.3. Decomposition of computational domain into eight subdomains.

subdomains in the *x*-direction. The figure is in the *x*-*y* plane showing only one layer of cells. All layers of cells in the *z*-direction will follow the same communication procedure.

The communications required in view of the dependencies in the equation for $\overline{u}^{yzt}$ are discussed first. From Fig 4.4 it can be seen that the variable $\overline{u}^{yzt}$ for the cells at the interface of two processors (corresponding to two subdomains) next to each other in the *x*-direction needs to be exchanged between them. The surfaces shown in the red color on the left processor are the ghost surfaces, which receive $\overline{u}^{yzt}$ variable computed at the red surfaces in the right processor. This communication is shown by the red (solid line) arrow. The green color surfaces and green (dashed line) arrow represent similar communication in the opposite direction. In addition, $\overline{u}^{xyz}$ from the cells at the interface is communicated from each processor to its neighboring processor in the negative *x*-direction. Since, $\overline{u}^{xyz}$ are node-averaged velocities, ghost cells (shaded cells in Fig. 4.4) are created to store the variable. The blue dots and blue (dotted line) arrow in the figure represent the above mentioned communication.

Considering the dependencies in equations of $\overline{u}^{zxt}$ and $\overline{u}^{xyt}$, similar communications are required in the *y* and *z* directions as well. It should be noted that the communications discussed above take care of the all the dependencies the $\overline{u}^{xyz}$ equation, except those arising from the pressure terms (arising through the f term). However, those communications are discussed later in the context of communications required for pressure equations.

The communications for corresponding variables in *v*, *w* and convection-diffusion equations are similar to those for the *u* equation.

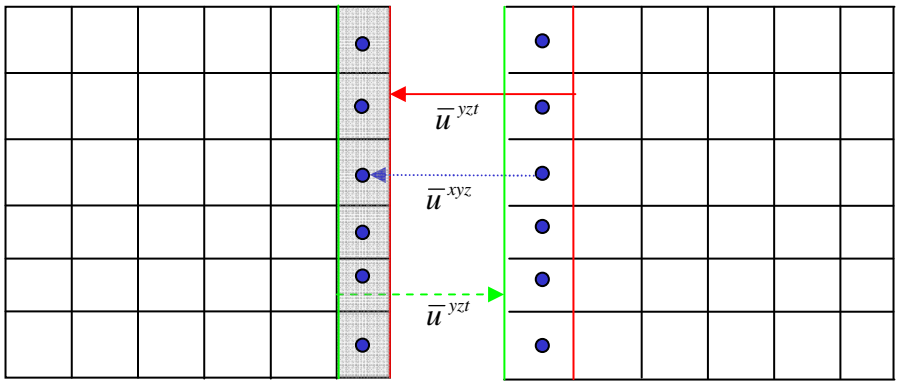

Fig. 4.4. Communications required for the velocity (or temperature) equations in the *x*-direction.

Figure 4.5, which is similar to Fig. 4.4, shows the communications of pressure variables between neighboring processors in the *x*–direction. From the dependencies in the equation of $\overline{p}^{yzt}$, it can be seen that $\overline{p}^{yzt}$ needs to be exchanged between the neighboring processors in the *x*-direction. This communication is shown by red (solid line) and green (dashed line) color arrows in Fig. 4.5. Also, $\overline{p}^{xyt}$, $\overline{p}^{zxt}$ and f_1 from the cells adjacent to the interface are communicated from each processor to its neighboring processor in the negative *x*-direction. The communications for $\overline{p}^{zxt}$ and f_1 are shown in brown (dash-dot line) and blue (dotted line) colors, respectively. The communication for $\overline{p}^{xyt}$ is not shown in the figure for sake of clarity. It should be noted that f_1 is derived from the velocity variables and can be computed locally once velocity variables are communicated. However, its computation requires exchange of several velocity variables which are not required for dependencies in velocity equations. Therefore, in order to reduce communication cost, f_1 is communicated rather than computing it locally. Similar communications are also required in the *y* and *z* directions respectively due to dependencies in the equations of $\overline{p}^{zxt}$ and $\overline{p}^{xyt}$. All the communications discussed so far are ideally needed at each iteration.

Special care is needed for communication of pressure variables in the corner cells. Fig. 4.6 shows the communication of $\overline{p}^{zxt}$ variable in the corner cell from where it is communicated to its destination. Though, this process is included in the communications described for pressure equations in the last paragraph, it is important to note that the communication shown by the red arrow (solid line) in the figure must precede the communication shown by the green arrow (dashed line). Otherwise, upper left processor

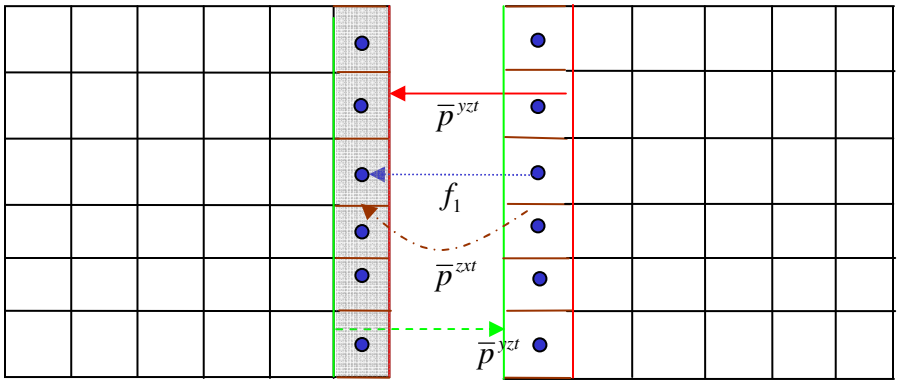

Fig. 4.5. Communications required for the pressure equations in the *x*-direction.

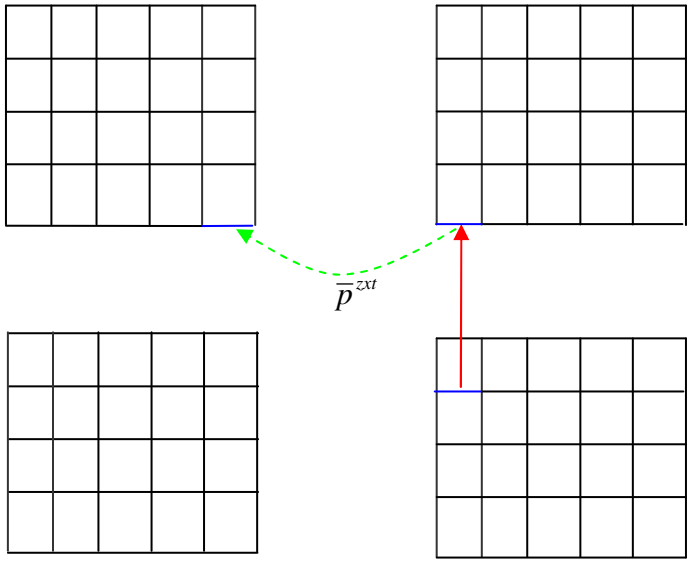

Fig. 4.6. Order of communication for the pressure variables in corner cell surfaces. Communication shown by solid arrow must precede that shown by dashed arrow.

in Fig. 4.6 will not have the correct value. Similar care in the order of communications is also required for $\overline{p}^{xyt}$ and $\overline{p}^{yzt}$ variables in the corner cells.

The *F* coefficients depend, in addition to the cell dimensions, on the previous time step velocities from that cell and the neighboring cells as discussed previously. Therefore, $\overline{u}_p$, $\overline{v}_p$ and $\overline{w}_p$ are also communicated to neighboring cells in the negative *x*, *y* and *z* directions, respectively, but only once at each time step. Since cell dimensions of the ghost cells are computed (in case of non-uniform grids) and stored locally, no additional communications are required.

4.2.3 Implementation using Message Passing Interface

The Message Passing Interface (MPI) (Gropp 1999a, Gropp 1999b) is used for parallelization of MNIM. MPI is a library specification for message-passing, proposed as a standard by a broadly based committee of vendors, implementers, and users. Various procedures of MPI used in parallelization of MNIM and their use in the implementation of the scheme are briefly discussed below:

MPI_DIMS_CREATE :- This procedure creates division of processors in a Cartesian grid depending on the number of dimensions given in the input. For example, in 3 dimensional topology, if 8 processors are assigned this procedure allocates two processors to each of the three directions. In case of 12 processors, the arrangement is 4×3×1.

MPI_CART_CREATE :- This procedure creates a new communicator with Cartesian topology based on the output of MPI_DIMS_CREATE. Each processor is assigned a coordinate according to its position in a virtual three dimensional array of the processors.

MPI_CART_SHIFT :- The procedure is used to identify immediate neighboring processors in all the directions. The procedure needs to be called once for each direction to assign neighbors in that direction. For periodic topology, the procedure identifies the neighbors suitably. The input argument of this procedure contains a logical array variable of size three to identify the directions in which the topology is periodic.

MPI_BCAST :- The input data is read by one of the processors and broadcasted to all the processors using this procedure. Before broadcasting, the input data is stored in two arrays, one for integer variables and for double precision variables. Only two calls of the procedure, one for each array, are then required to broadcast the data. Two arrays are required as two different *datatypes* cannot be communicated simultaneously. The values of various parameters are then assigned from those processors in each array.

It is pointed out here that one array can also be used to broadcast the input variables by using a derived *datatype* which consists of both *datatypes*. However, the broadcast is required only once and hence it is obviously not very useful to define a new *datatype* for the purpose.

Once the total number of cells is known to a processor, the starting and end cell numbers in each direction in that processor is evaluated depending on its position in the three dimensional array assigned by MPI_CART_CREATE. This is done so as to keep the computational load evenly balanced among the processors.

MPI_TYPE_VECTOR:- The communication procedures in the MPI are primarily built to communicate contiguous buffers of the same *datatype*. In the case of data being non-contiguous, a new MPI *datatype* can be defined using this procedure which then allows exchange of data without copying it into new arrays. The data exchanged in the *z*-

direction (this data has its indexes *i* and *j* varying and index *k* kept constant) is contiguous for the Fortran programming language. However, the exchanges in the other two directions require a new MPI *datatype* for each direction. The requirement of the two new *datatypes* is based on the fact that data non-contiguity is different in the *x* and *y* directions.

MPI_SEND_RECV:- During the iterative process the required communications, as discussed in the previous section, are carried out using this procedure.

MPI_ALL_REDUCE:- Each processor checks convergence after each Gauss-Seidel iteration only locally. Global convergence is checked by using this procedure.

4.2.4 Gathering of Variables

The velocity, pressure and temperature variables need to be collected together from all the processors after the completion of the computations for the purpose of post processing. The task is not straight forward because all the variables are stored in the local three dimensional arrays. Moreover, there are overlapping ghost cells created for the purpose of storing communicated variables.

There are several strategies which can be used for the abovementioned goal. MPI provides a procedure MPI_GATHER for collecting variables from all the processors to a single root processor. However, since data is non-contiguous because of overlapping cells, MPI also provides writing of output into separate files by individual processors. This approach, however, requires handling of several data files for post processing of the data. The use of MPI_DARRAY provides other alternatives to deal with the issue. The *darrays* are required to be used not only at the time of collection of variables but also in

the computation process. The process reduces the flexibility of the program because of the specific requirements of the *darrays*. In view of the above discussion, a separate approach is developed for the gathering of the variables.

The three dimensional arrays were unrolled into a one dimensional contiguous array. These contiguous arrays are then sent to a single processor (called root processor) by using the MPI_SEND command. In addition to that, information about start and end cell numbers in each direction is also communicated. The global three dimensional arrays are then constructed in the root processor. The approach developed here is not very efficient but is very reliable. Moreover, since the process is carried out only once at the end of the computations, the effect on overall efficiency of the program is negligible.

The approach, however, need to be modified for very large computations (involving thousands of processors) because single processor will not be able to handle all the data. Considerable amount of work is being carried out by parallel processing community to address this issue.

4.2.5 Communication Cost Relative to Computational Cost

The efficiency of parallel computation depends on efforts spent on communication between the processors relative to computations carried out in each processor. The efficiency is higher if the cost of the communication is less.

For a Gauss-Seidel solver, the computational cost per cell for MNIM is approximately four times that for finite volume method because there are four surface averaged variables for each velocity component. (The advantage of the MNIM results from the fact that the number of grid points, or cells, required is significantly less than

that required for finite volume method for the same level of accuracy (Wang 2003)). However, there are only two variables corresponding to each velocity component which need to be communicated between the processors. Therefore, in MNIM only half of the variables which are computed need to be communicated as opposed to all the variables in the case of finite volume method. Hence, it can be *a priori* concluded that parallelization of MNIM is likely to be more efficient than that for finite volume method. However, pressure equations need communications of four variables ($\overline{p}^{yzt}$, $\overline{p}^{zxt}$, $\overline{p}^{xyt}$ and f_1) in each direction leading to no advantage over finite difference or finite volume parallelization schemes. Since there are three velocity components and obviously only one pressure, overall parallelization efficiency of PMNIM is expected to be higher than that for finite difference or control volume schemes. However, direct comparison with other schemes is not possible unless exactly the same computing environment is used. Therefore, such studies have not been carried out.

In the above discussion the communication cost of previous time step velocities has been ignored as they are sent only once for each time step. This cost is assumed to be negligible compared with the communication cost of variables that must be transmitted at each iteration.

4.3 Numerical Results

4.3.1 Comparison with Benchmark Solution

The PMNIM is used to simulate natural convection in a three-dimensional cubic cavity of unit dimensions. The schematic diagram of the problem is shown in Fig. 4.7. The fluid in the cavity is air (Prandtl number = 0.71), and Rayleigh number is 10^4. Gravity is in the z-

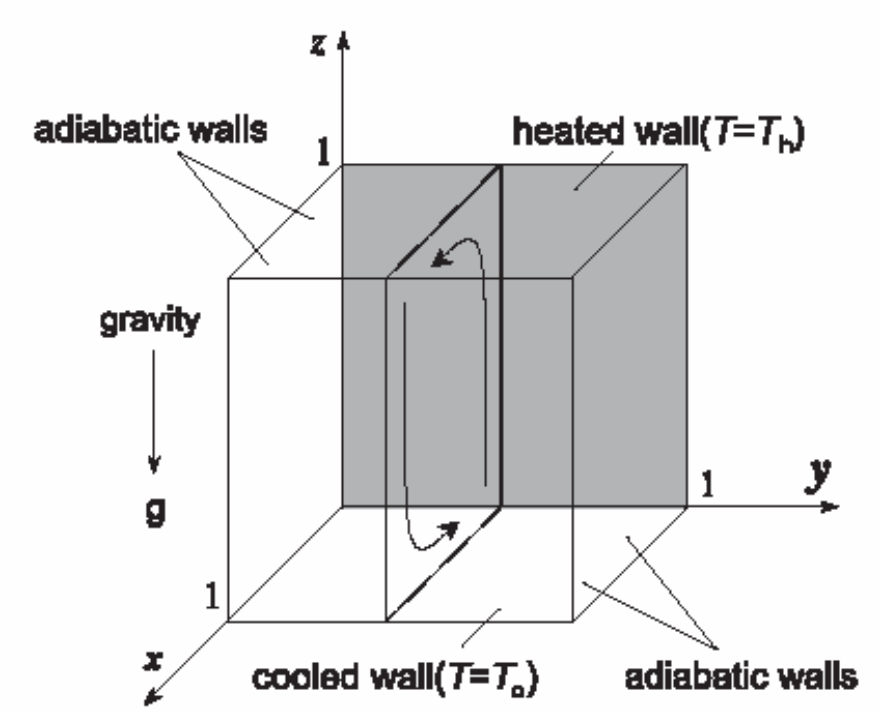

Fig. 4.7. Schematic of the test problem

direction. All surrounding walls are rigid and impermeable. The walls at $x = 0$ and $x = 1$ are isothermal but at different temperatures of T_h and T_c, respectively. The remaining four walls are adiabatic. The flow is assumed to be incompressible and laminar. The Boussinesq approximation has been assumed to be valid. The equations are solved in the primitive variables. Simulations are carried out on the Turing cluster which, at the time of these computations, consisted of 640 Apple Xserves, each with two 2 GHz G5 processors. The primary network connecting the cluster machines is a high-bandwidth, low-latency Myrinet network.

Figure 4.8 shows the velocity and temperature profiles at $y = 0.5$ plane for the test problem. Shown are the u and w velocity contour (a, b), isotherms(c), and vector plot of u-w velocities (d).

	PMNIM	Benchmark
u_{max} (z)	0.1995 (0.826)	0.1984 (0.825)
w_{max} (x)	0.2217 (0.117)	0.2216 (0.117)
Nu_{wall}	2.0621	2.0634
Nu_{center}	2.0751	2.0636

Table 4.1. Comparison of some of the results obtained using the PMNIM with the benchmark solution (Wakashima 2003)

Following quantities obtained using the PMNIM are compared in Table 4.1 with those reported for the benchmark solution obtained by Wakashima (2003):

u_{max}(z) : maximum center line ($x = 0.5$, $y = 0.5$) u-velocity (and its location).

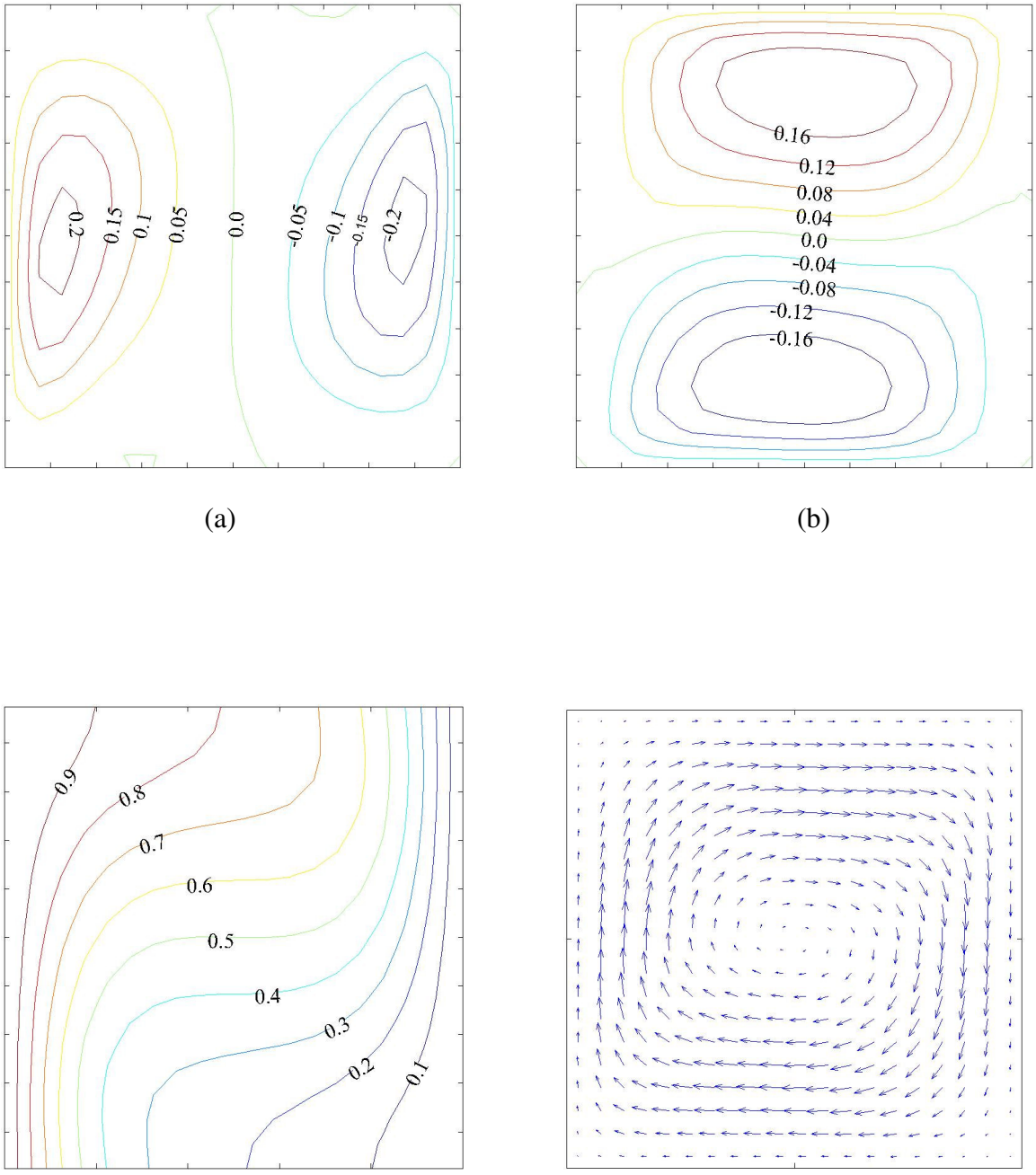

Fig. 4.8. Velocities and temperature at $y = 0.5$ plane (a) u–velocity contours, (b) w-velocity contours, (c) temperature contours, (d) vector plot of u-w velocities.

w_{max} (x) : maximum center line ($y = 0.5$, $z = 0.5$) w-velocity (and its location).

Nu_{wall} : Average Nusselt number over the hot (or cold) wall.

Nu_{center} : Average Nusselt number over the center plane ($x = 0.5$).

The average Nusselt number Nu over a plane parallel to the y-z plane is computed as follows:

$$Nu = \int_0^1 \int_0^1 \left(uT - \frac{\partial T}{\partial x} \right) dydz \tag{4.1}$$

The numerical results are *reliable*, i.e. results obtained using the parallel code with different number of processors are the same as those obtained using the serial code. Moreover, as can be seen from Table 4.1, results obtained with $20 \times 20 \times 20$ uniform grid compare well with the benchmark solution (Wakashima 2003). It should be noted that the benchmark solution is obtained on a uniformly distributed $80 \times 80 \times 80$ grid using a fourth order space time finite-difference method.

4.3.2 Efficiency with Fixed Problem Size Per Processor

It is now well established that the parallelization efficiency is best measured by increasing the problem size as the number of processors is increased (Gustafson 1988, Gustafson 1990). Therefore, parallelization performance is evaluated in this section by keeping the problem size for each processor fixed. In order to achieve this goal, the number of grid points or cells in each processor is kept constant as the number of processors is increased. However, as the total number of grid points or cells changes in

this case, number of iterations to converge will also change. Therefore, number of iterations is also kept fixed so as to achieve the abovementioned objective.

Since problem size per processor is fixed, therefore it is expected that the running time, in the absence of communication cost, should remain constant irrespective of the number of processors (known as Gustafson's law (Gustafson 1988)). In view of the above statement, speedup, *S*, for fixed problem size is defined as,

$$S = \frac{pT_s}{T_p} \tag{4.2}$$

and efficiency, *E*, is defined as

$$E = \frac{T_s}{T_p} \tag{4.3}$$

where p is the number of processors, T_s is runtime when a single processor is used and T_p is runtime when p number of processors are used.

The speedups and efficiency for a 10 × 10 × 10 grid in each processor are given in Table 4.2. The number of iterations performed is 5000. The computations take into account not only the effect of the number of processors but also the effect of processor configurations as well.

It is obvious that the efficiency of a given configuration depends on the processor with maximum number of neighbors. Moreover, if the highest number of neighbors in different configurations is the same, efficiencies are similar even if total number of processors is significantly different. For example, efficiency for 3 × 3 × 3 configuration is almost the same as that for 4 × 4 × 4 configuration because processors with maximum number of neighbors have 6 neighbors in both cases. The small decrease in efficiency can

be attributed to the fact that the communication cost associated with the convergence check depends only on the total number of processors and not on the number of grid points. Since the maximum number of neighbors will not exceed 6, therefore, the efficiency approaches a (nearly) constant value as the number of processors is increased. In other words, scalability of the scheme is quite good.

Processor Configuration	Time (s)	Speedup	Efficiency (%)
1×1×1	72.52	1	100
2×1×1	78.22	1.85	92.70
3×1×1	80.97	2.69	89.56
4×1×1	81.84	3.54	88.61
2×2×1	86.70	3.35	83.65
3×2×1	87.61	4.97	82.77
2×2×2	93.88	6.18	77.24
3×3×1	91.90	7.20	80.01
4×3×1	93.80	9.28	77.32
3×2×2	94.80	9.18	76.50
4×4×1	95.66	12.13	75.81
4×2×2	97.79	11.86	74.16
3×3×3	102.31	19.14	70.88
4×3×3	103.06	25.33	70.37
5×3×3	104.31	31.29	69.52
4×4×3	104.54	33.29	69.37
4×4×4	105.91	43.82	68.47

Table 4.2. Speedup and efficiency with $10 \times 10 \times 10$ grid for each processor for fixed problem size per processor.

It can be seen that the efficiency not only depends on the number of processors but also on the configuration of the processors i.e. the efficiency for the same number of processors is different for different processor configurations. Such a trend is expected since blocking communication MPI_SEND_RECV is used in the implementation of the

scheme. This behavior can be explained by considering the case of 2 × 2 × 1 configuration and comparing it with the case of 4 × 1 × 1 case. In 2 × 2 × 1 configuration each processor has to communicate with 2 neighboring processors in different directions. Although some processors in 4 × 1 × 1 configuration do have to communicate with 2 processors, the communication is only in the *x*-direction. Since MPI_SEND_RECV simultaneously communicates all information in one direction (say, *x*) and only then starts communicating in the other directions, the result is that the efficiency of 4 × 1 × 1 configuration is higher among the two cases considered here. Moreover, 4 × 1 × 1 and 3 × 1 × 1 configuration efficiencies are not significantly different. Figure 4.9 shows the speedup for the cases in which at least one processor has 6 neighbors.

Processor Configuration	Time (s)	Speedup	Efficiency (%)
1×1×1	2542.66	1	100
2×1×1	2846.66	1.79	89.32
3×1×1	2877.33	2.65	88.36
4×1×1	2899.66	3.50	87.68
2×2×1	2900.33	3.47	87.66
3×2×1	2900.66	5.26	87.65
2×2×2	3050.33	6.67	83.35
3×3×1	2967.01	7.71	85.69
4×3×1	2985.33	10.22	85.17
3×2×2	3000.66	10.09	84.05
4×4×1	3025.00	13.55	84.73
4×2×2	3038.66	13.39	83.67
3×3×3	3291.05	20.86	77.26
4×3×3	3258.66	28.09	78.02
5×3×3	3364.03	34.01	75.58
4×4×3	3346.02	36.47	75.99
4×4×4	3300.05	49.31	76.05

Table 4.3. Speedup and efficiency with 20 × 20 × 20 grid for each processor for fixed problem size per processor.

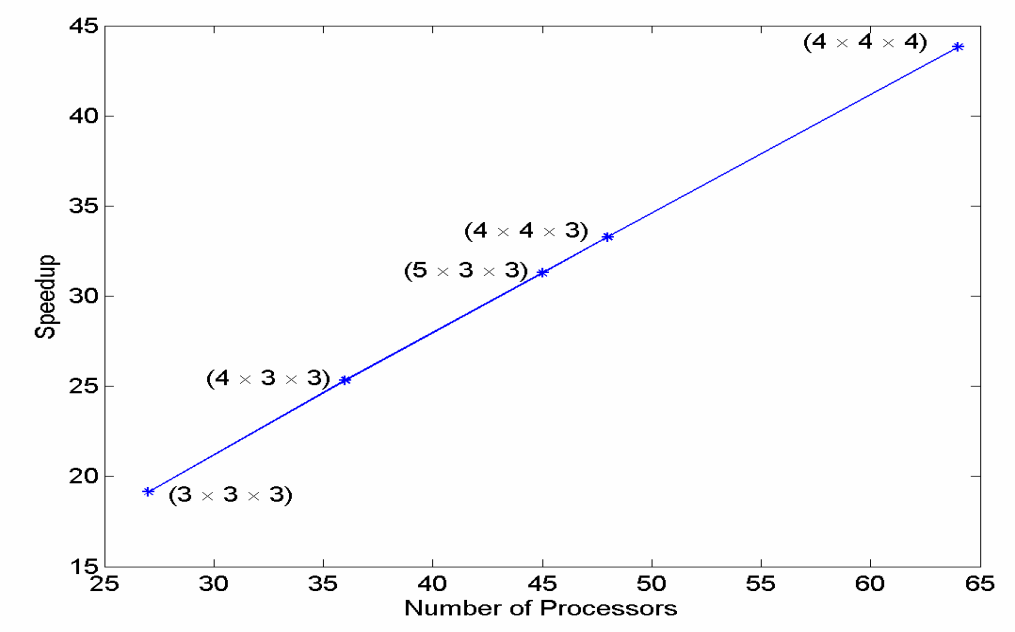

Fig. 4.9. Speedup of PMNIM with $10 \times 10 \times 10$ grid for fixed problem size per processor. All cases shown here have at least one processor with 6 neighbors. Efficiency ranges from 68.47% to 70.88%

The efficiencies with different number of processors for the 20 × 20 × 20 grid (for 5000 iterations) are given in Table 4.3. The trends observed for the 20 × 20 × 20 grid case are similar to those for the 10 × 10 × 10 grid. The ratio of cells at the interface to total number of cells decreases as total number of grid points is increased. Therefore, in general, efficiency of 20 × 20 × 20 case should be higher than 10 × 10 × 10 grid. However, this is true only if communication cost of sending data is independent of the size of the data. This assumption is usually not valid because of limitations of the bus size used for data exchange. For the 3D thermal cavity problem solved using the PMNIM, however, the efficiency of the 20 × 20 × 20 case for some configurations is lower than the 10 × 10 × 10 grid case while it is higher for other configurations. The result may be explained by the fact that contiguity of data to be sent in different directions is not the same. This result in higher communication cost in the direction in which communicated data is non-contiguous. Figure 4.10 shows the speedup for the cases in which at least one processor has 6 neighbors.

4.3.3 Efficiency with Variable Problem Size Per Processor

In this sub-section, while the problem size per processor is varied, the overall problem size is kept fixed, i.e. the *total* number of cells is kept constant. The problem size per processor, therefore, decreases as number of processors increase. The number of iterations remains almost the same irrespective of the number of processors used. The definition of speedup, *S*, relevant for this sub-section, for a given number of processors, is as follows:

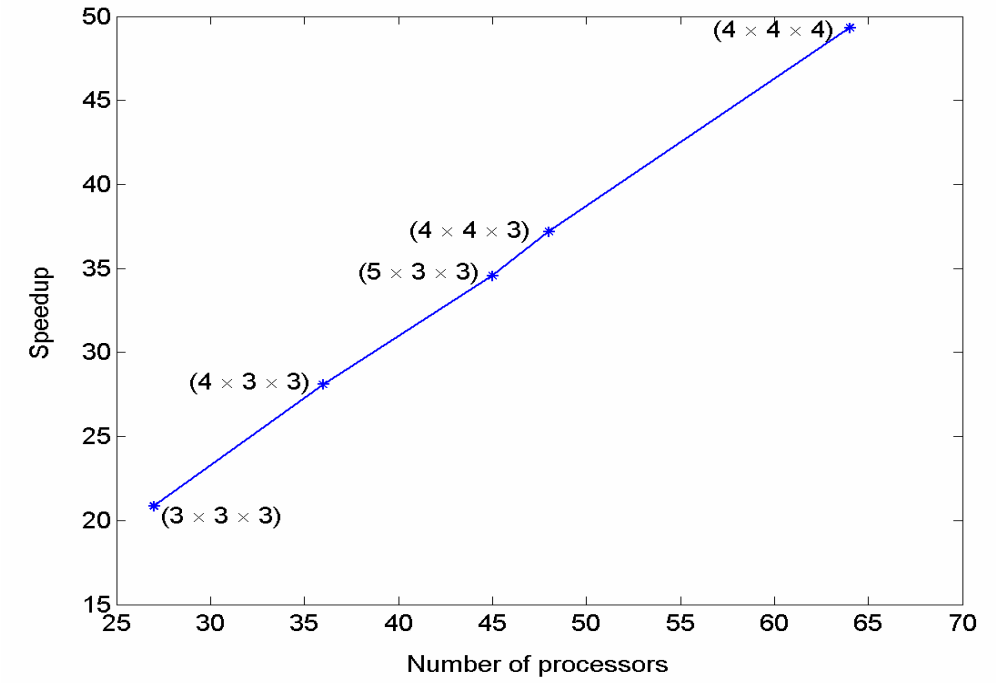

Fig. 4.10. Speedup of PMNIM with 20 × 20 × 20 grid for fixed problem size per processor. All cases shown here have at least one processor with 6 neighbors. Efficiency ranges from 75.58% to 78.02%.

$$S = \frac{T_s}{T_p} \tag{4.4}$$

The efficiency E is defined as:

$$E = \frac{T_s}{pT_p} \tag{4.5}$$

The performance is evaluated for two cases with 12 × 12 × 12 and 20 × 20 × 20 grid. Speedup and efficiency for PMNIM with 12 × 12 × 12 grid are presented in Table 4.4. Figure 4.11 shows the plot for the speedup for 12 × 12 × 12 grid case. It can be seen that as the number of processors is increased the speedup and efficiency decrease. This expected behavior is observed because, for a given grid, the communication cost goes up with the number of processors while computation cost remains unchanged. This behavior is known as Amdahl's law (Amdahl 1967). It can be seen from Table 4.4 that the drop in efficiency is rapid for relatively higher number of processors. This rapid drop is due to the fact that a small number of cells are being distributed over a large number of processors resulting in large number of cells at interfaces, which need communication of variables, thereby increasing the communication cost relative to the computation cost. For example, with 24 processors number of cells in each processor is 72 while number of cells at interfaces is 96, leading to reduced efficiency.

Table 4.5 presents the speedup and efficiency of PMNIM with 20 × 20 × 20 grid. Figure 4.12 shows plot for the speedup for 20 × 20 × 20 grid case. The efficiency and speedup trends for this case are very different from the 12 × 12 × 12 grid case. The efficiency increases with increasing number of processors initially and only then starts to decrease. Moreover, efficiency is greater than 100%. Superlinear efficiencies have been encountered in the past and they are associated with memory/cache effects (Gustafson

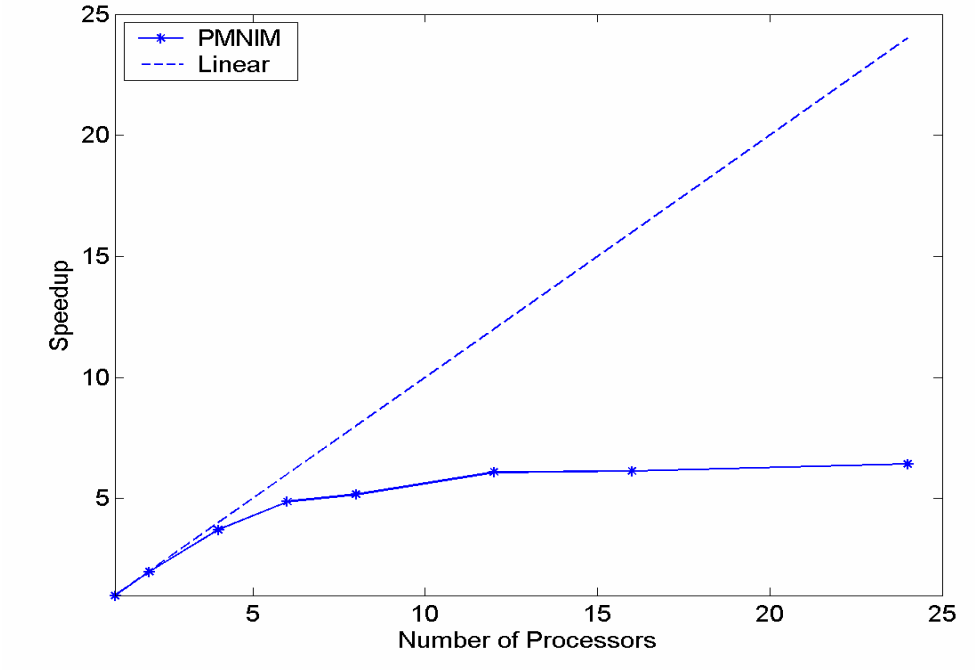

Fig. 4.11. Speedup of PMNIM with 12 × 12 × 12 grid for variable problem size per processor.

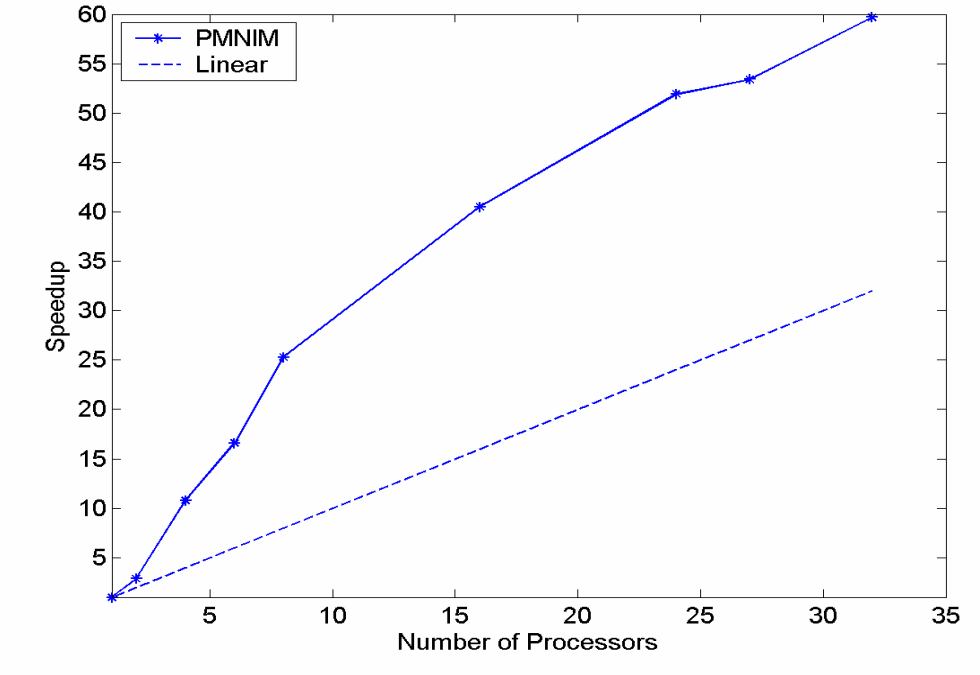

Fig. 4.12. Speedup of PMNIM with $20 \times 20 \times 20$ grid for variable problem size per processor.

1990, Helmbold 1989, Parkinson 1986). Therefore, it is possible that for a relatively large problem, the memory required exceeds the size of the faster memory type (cache) if a small number of processors is used. This results in increase in efficiency with number of processors because a larger fraction of the memory need is satisfied by the faster memory type as the number of processors is increased (Gustafson 1990, Helmbold 1989, Parkinson 1986). The efficiency starts decreasing only when the communication cost starts dominating over the memory effect. CPU times for different number of processors in Table 4.5 suggest that a qualitatively different kind of change takes place as the number of processors is changed from 6 to 8, suggesting that for given capabilities of the processors used and the problem size (20 × 20 × 20), minimum of 8 processors are needed to satisfy the memory requirement entirely by the faster memory.

Processors	Time (m)	Speedup	Efficiency (%)
24	2.77	6.42	26.8
16	2.90	6.13	38.3
12	2.92	6.09	50.7
8	3.43	5.17	64.6
6	3.65	4.87	81.2
4	4.78	3.71	92.8
2	9.03	1.97	98.6
1	17.78	1	100.0

Table 4.4. Speedup and efficiency with 12 × 12 × 12 grid, for variable problem size per processor.

In such cases an alternate definition of speedup and efficiency may be based on the CPU time corresponding to the minimum number of processors, *m*, that allow the

problem to fit in cache, as the normalizing factor. Hence speedup, *S*, and efficiency, *E*, may be defined as

$$S = \frac{T_m}{T_p}; \text{ for } p \geq m \tag{4.6}$$

$$E = \frac{mT_m}{pT_p}; \text{ for } p \geq m \tag{4.7}$$

Processors	Time (m)	Speedup	Efficiency (%)
32	14.6	59.7	186
27	16.3	53.4	197
24	16.8	51.9	216
16	21.5	40.5	253
8	34.5	25.3	316
6	52.4	16.6	276
4	80.4	10.8	270
2	304.2	2.86	143
1	871.3	1	100

Table 4.5. Speedup and efficiency with 20 × 20 × 20 grid,

for variable problem size per processor

where T_m is time corresponding to *m* processors. For data given in Table 4.5 (for 20 × 20 × 20 grid case) this re-normalized speedup is plotted in Fig. 4.13. Extracting the cache effect from the definition of speedup clearly leads to speedups that are *sublinear*. [Here, it

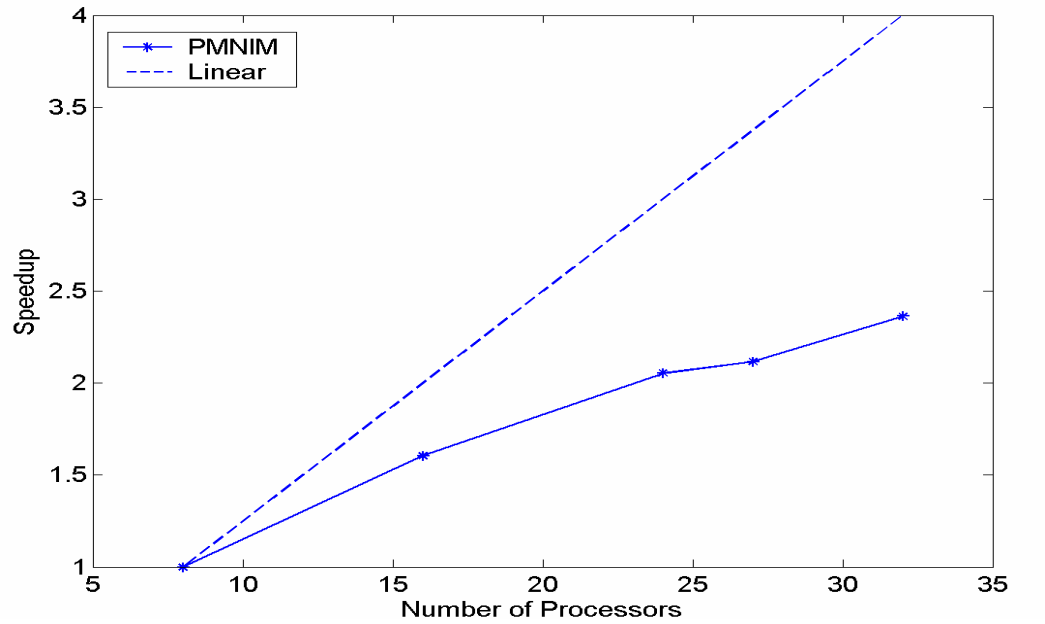

Fig. 4.13. Speedup of PMNIM with 20 × 20 × 20 grid for variable problem size per processor. Here all CPU times are normalized using the CPU time for eight processor case

should be noted that for same number of grid points or cells, memory requirements for MNIM are much higher than many other numerical schemes. Memory requirements for MNIM even for smaller number of cells needed to achieve accuracy comparable to that obtained with much finer grids with other conventional schemes is likely to be relatively higher. This is due to the fact that four variables are computed for each velocity component per cell instead of one in most other schemes. Similarly, number of variables for pressure and temperature is also higher. Moreover, coefficients that appear in the discrete equations are stored to avoid repeated evaluation. Therefore, the memory effects are significant for PMNIM even for comparatively small number of cells in the problem.]

4.4 Summary of Parallelization of MNIM

Parallel version of MNIM has been developed and tested. The domain decomposition paradigm used for implementation of PMNIM, though commonly used for grid based (e.g. finite volume) methods, requires a unique approach because of a different and more complex template in the MNIM when compared with most other commonly used schemes. The speedup and efficiency of PMNIM has been studied with its application to the problem of natural convection in a cubic cavity.

For a fixed problem size, the efficiency primarily depends on the communication cost of the processor with largest number of processors and to a much smaller extent on the total number of processors. The efficiency is nearly the same for the cases where at least one processor has 6 neighbors, irrespective of the total number of processors used, showing that the PMNIM is scalable. Moreover, considering that relatively small number of grid points have been used for performance evaluation, the scheme shows the potential

to be quite efficient as well. For a variable size problem, the efficiency of PMNIM decreases with increasing number of processors for small number of grid points. This behavior is expected and commonly observed in parallel applications because communication cost relative to computation cost increases with increasing number of processors. However, for relatively large number of grid points superlinear speedups are possible for PMNIM. Superlinear speedups result because of the large memory requirements of the MNIM that result in significant memory effects even for relatively small number of cells.

Chapter 5

Direct Numerical Simulation using Parallel Modified Nodal Integral Method

Direct Numerical Simulation (DNS) of flows is a relatively new tool for turbulence research. Though low resolution computations for isotropic turbulence were done at National Center for Atmospheric Research in 1972 by Fox (1972), the real progress in DNS started in 1980s. In the early part of that decade Rogallo (1981) studied the effects of the mean shear in homogenous turbulence using DNS. DNS was further used to compute free shear layers (Riley 1986). DNS of wall bounded flows started in 1987, when plane channel flow and curved channel flow were simulated by Kim (1987) and Moser (1987), respectively. Since then DNS has progressed to more complex flows such as channel flow with rotation (e.g. Cambon 1997), heat transfer (e.g. Kasagi 1992) etc. Recent years have also seen the application of DNS to flows which are inhomogeneous in the streamwise direction (e.g. Le 1992) as well as to compressible flows (e.g. Huang 1995).

Recent progress in DNS has led to computation of relatively complex flows but their Reynolds numbers are still low. Therefore, application of DNS to engineering flows

will not be possible in the near future. In the meantime DNS is continued to be used as a research tool.

One very useful aspect of the DNS is that it can be used to simulate flows which are difficult or impossible to study experimentally. Such simulations are carried out, for example, to study flow response to isolated physical parameters. Several numerical experiments such as forced isotropic turbulence (Eswaran 1988) and minimal channel (Jimenez 1991) have been designed to study turbulent flows. These numerical experiments have led to significant insight into the physics of turbulence. Moreover, DNS is also complementary to experiments in turbulence research. For example, Suzuki (1992) used DNS results from a turbulent channel flow to quantify errors in hot wire measurements.

Another significant contribution of the DNS has been in the field of turbulence modeling. The Reynolds stress equations, which form the basis for closure of Reynolds averaged mean flow equations, have several terms which must be modeled. These terms are difficult to measure in experiments. However, using DNS all terms can be computed directly. Therefore, any simplified representation of a term (or physical process) can be directly tested by comparing it to the DNS results.

The DNS is also quite successful in simulating transition to turbulence in flows. Due to relatively low Reynolds numbers in these flows (when compared to fully turbulent flows), the flow simulation is relatively less computationally intensive for relatively complex flows. An early review of the numerical simulation of the abovementioned transition can be found in the review by Kleiser (1991). The recent work in this area has been reviewed by Schmid (2001).

5.1 Numerical Issues

The range of length scales and time scales is large in turbulent flows and it also increases rapidly with Reynolds number. Numerical methods for DNS are expected to accurately represent wide range of scales. The largest scales in the inhomogeneous direction are determined by physical parameters (e.g. channel width). In the homogenous direction the length scales are of the order of the integral scale. The smallest scales resolved need to be of O(η) where η is the Kolmogorov length scale given by $\eta = (\nu^3 / \varepsilon)^{1/4}$. The resolution requirement will obviously depend on numerical scheme used.

In order to resolve the whole range of the length scales, the computational resources required are proportional to the ratio of these large and small scales. This ratio is large and increases as cube of the Reynolds number. Therefore, extensive computational resources are required for the DNS especially at large Reynolds numbers. The focus of current work, because of limited availability of resources, is on the stability and transition to turbulence in some flows. Specifically, the PMNIM has been used to simulate the transition to turbulence in the ABC (Arnold-Beltrami-Childress) flows. The details of the ABC flow are given in the subsequent sections.

5.2 ABC Flows

ABC flow is the solution of the 3D Navier-Stokes equations:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}.\nabla\mathbf{v} = -\nabla p + \frac{1}{R}\nabla^2\mathbf{v} + \frac{\mathbf{F}}{R} \tag{5.1}$$

where $\mathbf{v} = (u,\ v, w)$ and $\mathbf{F} = (F_x, F_y, F_z)$ is defined as

$$\frac{\mathbf{F}}{k^2} = (\,A\sin(kz) + C\cos(ky),\ B\sin(kx) + A\cos(kz),\ C\sin(ky) + B\cos(kx)\,) \tag{5.2}$$

Equation (5.1) is solved along with the continuity equation:

$$\nabla.\mathbf{v} = 0 \tag{5.3}$$

The periodic boundary conditions are assumed to be applicable, which are given as follows,

$$\begin{aligned} \mathbf{v} &= \mathbf{v}(\ x + 2p\pi,\ y + 2q\pi,\ z + 2r\pi\ ) \\ p &= p(\ x + 2p\pi,\ y + 2q\pi,\ z + 2r\pi\ ) \end{aligned} \tag{5.4}$$

where, p, q and r can be any three integers.

For small values of R, the above set of equations has the following trivial (laminar) solution:

$$\mathbf{v} = (\ A\sin(kz) + C\cos(ky),\ B\sin(kx) + A\cos(kz),\ C\sin(ky) + B\cos(kx)\ ) \tag{5.5}$$

It should be noted that the ABC flow is a Beltrami flow i.e. it satisfies

$$\nabla \times \mathbf{u} = \alpha\mathbf{u} \tag{5.6}$$

with $\alpha = 1$. It should also be noted that Eq. (5.5) is also the solution of the following force free Euler equation,

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}.\nabla\mathbf{v} = -\nabla p \tag{5.7}$$

Arnold (1965) has shown that force free Euler equation can have chaotic streamlines only if it is a Beltrami flow and ABC flow is one of the simplest flows possessing this property. For this reason, this flow has been extensively studied by mathematicians and

physicists. Henon (1966) has shown numerically, for a particular set of values of A, B and C, that the flow is indeed chaotic. Extensive study of this flow has been carried out by Dombre (1986). In fact, the name "ABC Flow" has been first used in this work.

The ABC flows are also studied in the context of the kinetic dynamo problem. In this context, ABC flows were first studied by Childress (1970). It has also been shown in various works that these flows act as fast dynamos (e.g. Galloway 1984, Gilbert 1992).

There have been some relatively recent studies which consider the ABC flows as prototype for the development of turbulence. The work on the stability of the ABC flows, in this context is carried out by Podvigina (1994). Some recent work on the stability and bifurcation in this area by Podvigina (1999), Ashwin (2003) and Podvigina (2006) are also notable.

In the present work, the ABC flow equations are solved for,

$$A = B = C = 1 \tag{5.8}$$

For the above values of the coefficients it is called 1:1:1 ABC flow.

5.2.1 Heteroclinic Cycles in the ABC Flows

The heteroclinic cycles are defined as a collection of solution trajectories that connect sequences of fixed points. The fixed points can be equilibria, periodic solutions or chaotic invariant sets. Such cycles are characterized by metastable, recurrent behavior, made up of long periods of quasi-static regimes with sudden bursts of aperiodic, spatio-temporal evolution, relaxing after a while to new quasi-static states (Lloyd 2005). These cycles are observed in systems which possess a symmetry group with large number of elements. Spontaneous or forced symmetry breaking, which reduces the number of these

elements, generates heteroclinic cycles. The heteroclinic cycles have been seen in the solutions of various PDEs including the Navier-Stokes equation. The description of the heteroclinic cycles are given by Melbourne (1989), Krupa (1995) and Field (1996). A survey of the heteroclinic cycles is given by Krupa (1997).

The occurrence of the heteroclinic cycles is relatively rare in practical flows because of the absence of large number of elements in the symmetry group of such flows. However, there are some flows where the heteroclinic cycles can be seen, for example, in Rayleigh Bernard convection (Mercader 2001). The heteroclinic cycles are seen in the ABC flows due to the large number of the elements in the symmetry group in the governing equations of these flows.

5.2.2 Symmetries in the ABC Flows

The symmetry group for the ABC flow is generated by the following symmetries:

$$\begin{aligned}(x, y, z) &\rightarrow (\pm x, \pm y, \pm z)\\ (x, y, z) &\rightarrow (y, z, x)\end{aligned} \tag{5.9}$$

The superposition of the above symmetries generates a discrete symmetry group of 24 elements for the ABC flows. It is obvious that the symmetry group is independent of the value of k in the ABC flow equations. It should be noted that the time reversal symmetry ($t \rightarrow -t$) is not considered in the above group because it is immaterial for the generation of heteroclinic cycles. The heteroclinic cycles are generated by the spontaneous symmetry breakdown due to non-linear instabilities at large R.

5.2.3 ABC Flow as a Test Problem

The reason ABC flow is selected as a test problem for PMNIM is that, unlike isotropic turbulence, the initial conditions and results are available in the literature in real space and not in the Fourier space. Moreover, the demand on the spatial grid refinement for the ABC flow is not as severe as it is for other similar Reynolds number turbulent flows resulting in lesser demand on the computational resources. However, it needs to be mentioned, that very large integration times are needed to study heteroclinic cycles arising in these flows. Therefore, even with relatively coarse grids, the simulation of these flows is quite computationally intensive.

ABC flow is generally studied using spectral methods. Some of the work on the instabilities of this flow has been reproduced here using PMNIM. The results are compared with the results obtained using spectral methods reported by Podvigina (1994) and Podvigina (2006). These results are obtained with $k = 1$ in Eq.(4). In addition to these simulations, the computations are also carried out for the $k = 2$ case.

5.3 DNS of ABC Flows using PMNIM

To study the ABC flows using the PMNIM formulation developed in the previous chapter the force term has to be evaluated in accordance with the PMNIM formulation. It follows from Eq. (2.26), that to evaluate $f_{i,j,k,n}$ in a cell, the average of the force term is needed over that space-time cell. It should be noted that this average needs to be calculated corresponding to the force term in each direction. This term can be evaluated for the force term in the x - direction as follows:

$$\overline{F}^{xyzt}_{x\ i,j,k,n} = \frac{1}{16 a_i b_j c_k \tau_n} \int_{t_n-\tau_n}^{t_n+\tau_n} \int_{z_k-c_k}^{z_k+c_k} \int_{y_j-b_j}^{y_j+b_j} \int_{x_i-a_i}^{x_i+a_i} k^2 \begin{pmatrix} A\sin(kz) \\ +C\cos(ky) \end{pmatrix} dxdydzdt \quad (5.9)$$

where, x_i, y_j, z_k and t_n are the coordinates of the mid point of the space-time cell. The above integral yields,

$$\overline{F}^{xyzt}_{x\ i,j,k,n} = k^2 \left(\frac{A\sin(kz_k)\sin(kc_k)}{kc_k} + \frac{B\cos(ky_j)\sin(kb_j)}{kb_j} \right) \quad (5.10)$$

Note that the averaged force term reduces to the point value at the mid point of the cell as cell size decreases. The averaged force term in the other directions can be similarly obtained.

It should also be noted that all the derivatives of the forcing term, appearing in the pressure Poisson equation, are identically zero (see Eqs. (2.33) and (2.34)). Therefore pressure equation need not be modified for the simulation of ABC flows.

5.4 Results for $k = 1$ case

The ABC flow described in the preceding sections has been previously solved by Podvigina (1994), for $k = 1$, using the Fourier pseudo spectral method (FPSM). Third and fourth order Runge-Kutta schemes were used with time steps of 0.01. The grid sizes used were 16^3 and 32^3 for the lower and higher values of R, respectively. In the present work, the governing equations are solved for the same flow with the grid size of 20^3 and the time step is kept at 0.015 for all the simulations. All the FPSM results shown in this dissertation are either from Podvigina (1994) or Podvigina (2006). It is also pointed out that Podvigina (1994) used the trivial solution that was slightly perturbed symmetrically,

as an initial condition. The perturbations were applied symmetrically in the Fourier modes. However, no explicit perturbations are applied in the present simulations, and the only perturbations present are from the inherent numerical error. Since, PMNIM solves the equations in the real space; the perturbation similar to those used by Podvigina (1994) can not be applied. The initial condition for all the simulation is one of the solutions from the secondary set of the solutions which are described later in this chapter.

The only solution for small R is the trivial solution given by Eq. (5.2). Figure 5.1 (a) shows the plot of the u-velocity as a function of z, evaluated using PMNIM for a small value of R (= 10). The velocity is plotted at ($x = \pi$, $y = \pi$). Corresponding exact solution at that location is also plotted in the same figure. Figure 5.1 (b) is the plot of the same component of the velocity as a function of y at ($x = \pi$, $z = \pi$). The exact solution is also plotted for comparison. It should be noted that the u-velocity is independent of x in the whole domain. Figure 5.2 shows plots similar to those in Fig. 5.1, for the v-velocity. The corresponding exact solutions are also plotted. The plots show very good agreement with the exact solution. The plots for the w-velocity are similar and are not shown here for the sake of brevity.

At higher values of R, a secondary set of three solutions also exist, one of which is given by

$$\mathbf{v} = (\ \tilde{A}\sin(kz) + \tilde{C}\cos(ky),\ \tilde{B}\sin(kx) + \tilde{A}\cos(kz),\ \tilde{C}\sin(ky) + \tilde{B}\cos(kx)\ ) \tag{5.11}$$

where,

$$\tilde{A} = \tilde{B} \approx 0.48,\ \tilde{C} \approx 1.34,\ k = 1 \tag{5.12}$$

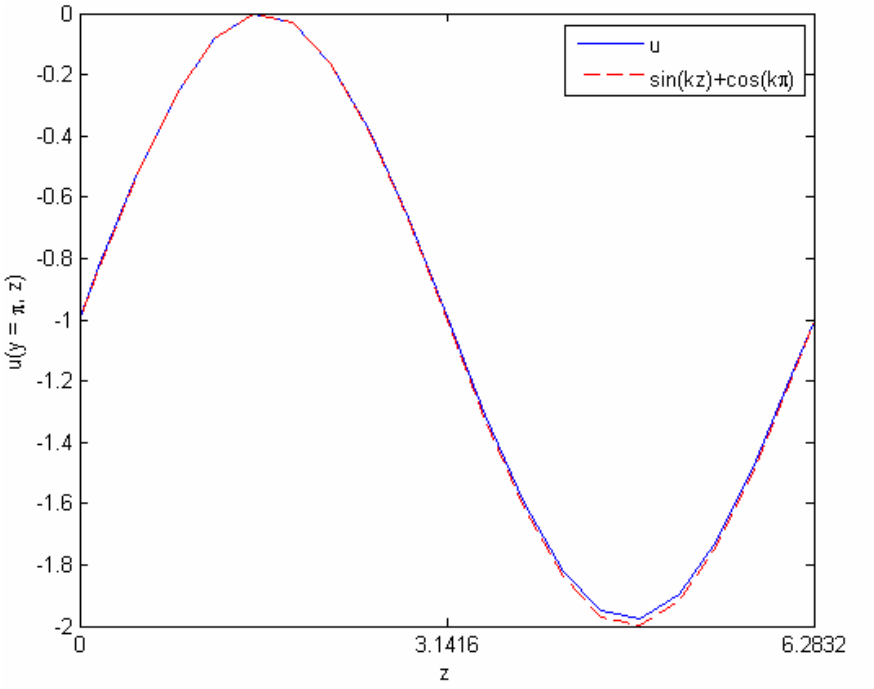

Fig. 5.1 (a) Plot of the u - component of the velocity for $R = 10$ as a function of z. The velocity is plotted at $y = \boldsymbol{\pi}$.

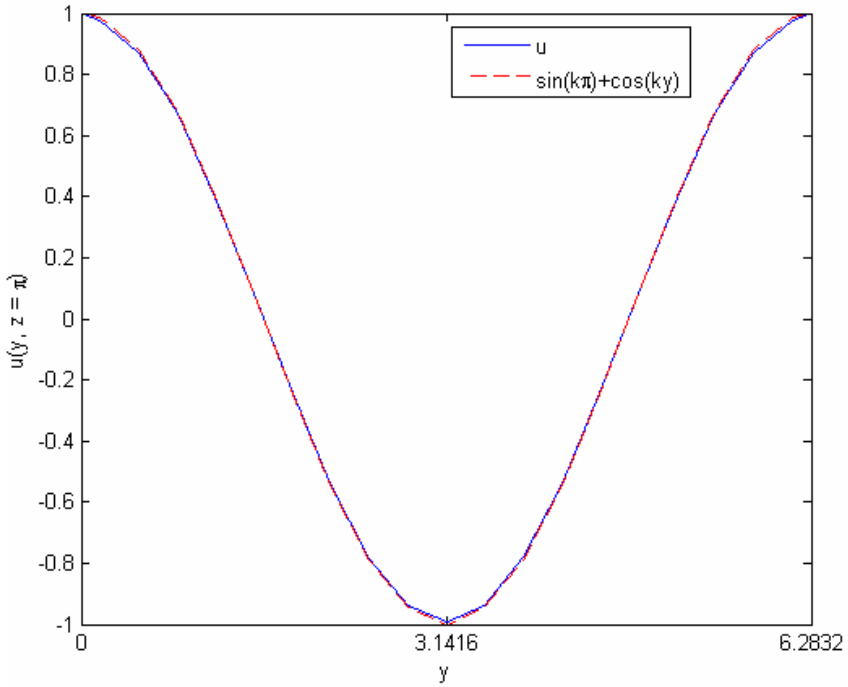

Fig. 5.1 (b) Plot of the u - component of the velocity for $R = 10$ as a function of z. The velocity is plotted at $z = \boldsymbol{\pi}$.

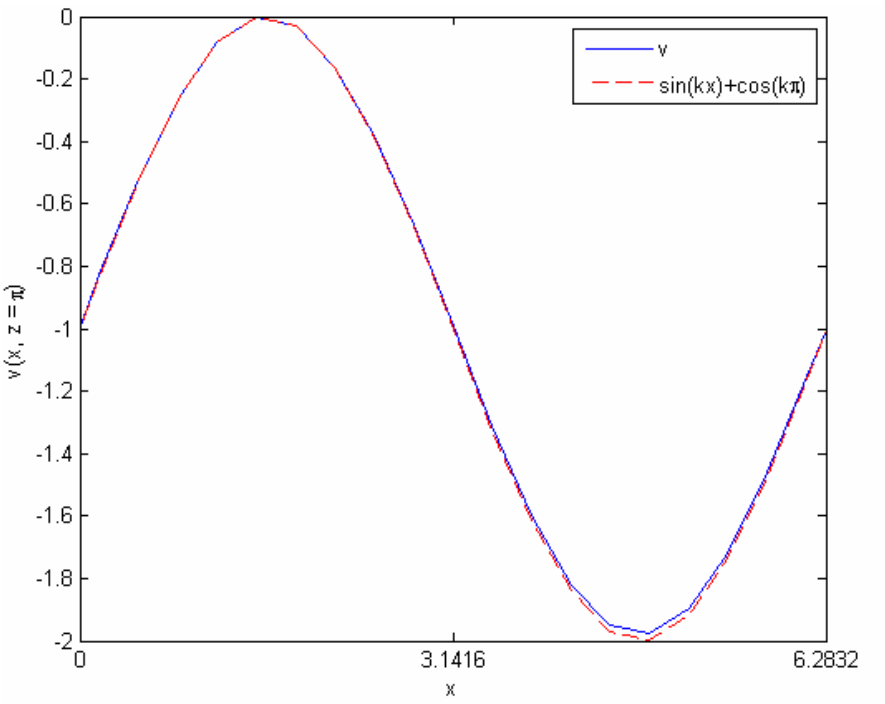

Fig. 5.2 (a) Plot of the v - component of the velocity for $R = 10$ as a function of x. The velocity is plotted at z = $\boldsymbol{\pi}$.

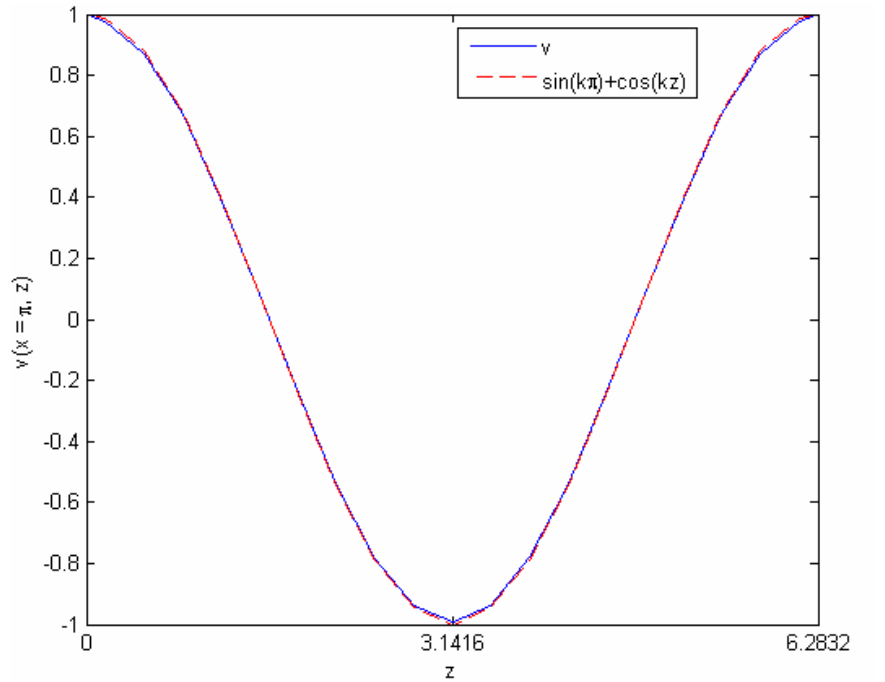

Fig. 5.2 (b) Plot of the v - component of the velocity for $R = 10$ as a function of z. The velocity is plotted at $x = \boldsymbol{\pi}$.

Due to the symmetry in the flow, the solutions obtained by the permutations of the coefficients in Eq. (5.11) are also the solutions of the governing equations. The total energy of the three solutions is obviously the same for the three cases.

Before proceeding further with the discussion of the results, two quantities which are plotted in the subsequent figures need to be defined. First is the average energy in the entire domain, which is defined as,

$$E = \frac{1}{2V}\int_V (\mathbf{v}.\mathbf{v}) dV \,. \tag{5.13}$$

The second is the energy contained in any one component of the velocity, averaged over the entire domain, defined as,

$$E_n = \frac{1}{V}\int_V v_n^2 dV \,. \tag{5.14}$$

where, n = 1, 2 and 3 represent the velocities in the x, y, and z directions, respectively. In the above two equations V is the volume of the computational domain. It should be noted that for the trivial solution (Eq. (5.5)) the average energy is 1.5 and E_n of each of the component is 1. For the case where Eq. (5.11) is valid the average energy is lower than the trivial case. Moreover, E_n of two of the components is the same; however, E_n of the third component is higher than that of the other two.

In Fig 5.3, the average energy of the three components of the velocity is plotted as a function of time. The plot is for a low value of R (= 10) and the initial condition is the solution given by the Eq. (5.11). The figure shows that, at low values of R, the solution of the equations evolves to the trivial solution if started with an initial condition given by Eq. (5.11). In contrast, the solution returns to the initial condition for relatively larger

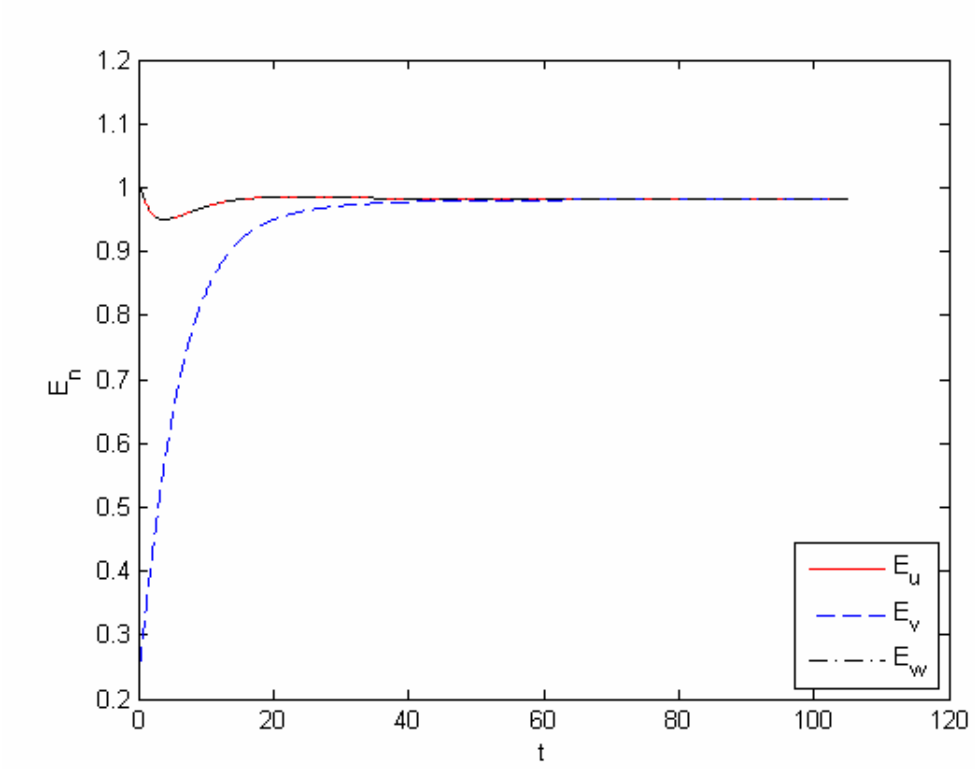

Fig. 5.3 Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 10) obtained using PMNIM. The initial condition is one of the secondary solutions which evolve into trivial solution with increasing time.

values of R, as shown in the Fig. 5.4. This plot is obtained for $R = 11$. It should be noted that it has been shown theoretically that both trivial and secondary steady state solution can exist for $0.5 < R < 13$ (Podvigina 1994). However, as seen from the plots, the trivial solution in the present case is obviously a stronger attractor for $R < 10$. It is also pointed out that works by Podvigina (1994) and Podvigina (2006) focuse only on R greater than 13 and hence PMNIM and FPSM can not be compared for the above analysis.

Figures 5.5 and 5.6 show the plots of the velocities for $R = 11$ obtained using PMNIM. These figures are similar to Figs 5.1 and 5.2, respectively, although, at a different value of R. Therefore, solutions are plotted with the exact solutions given by Eq. (5.11), which is the valid solution at this value of R. The plots show that the solutions obtained using PMNIM match well with the exact solutions.

The trivial steady state become unstable at $R \sim 13$ and shows Hopf type of bifurcation. In the present research, this and several of the other results obtained using PMNIM are found to be similar to the results given by Podvigina (1994). However, only a few salient features are compared here for the sake of brevity, especially considering the fact that the main goal here is to test the ability of the PMNIM to accurately simulate chaotic flows.

The Fig. 5.7 shows the evolution of E with time at $R = 14.9$. The results are similar to those obtained by FPSM at $R = 14.6$ as shown in the same figure. The figure shows the Hopf type bifurcation in the secondary steady state solution, resulting in a periodic solution. Figure 5.8 shows the solution at $R = 15.5$ obtained using PMNIM and the available FPSM solution at $R = 15.3$. These solutions show a heteroclinic cycle which

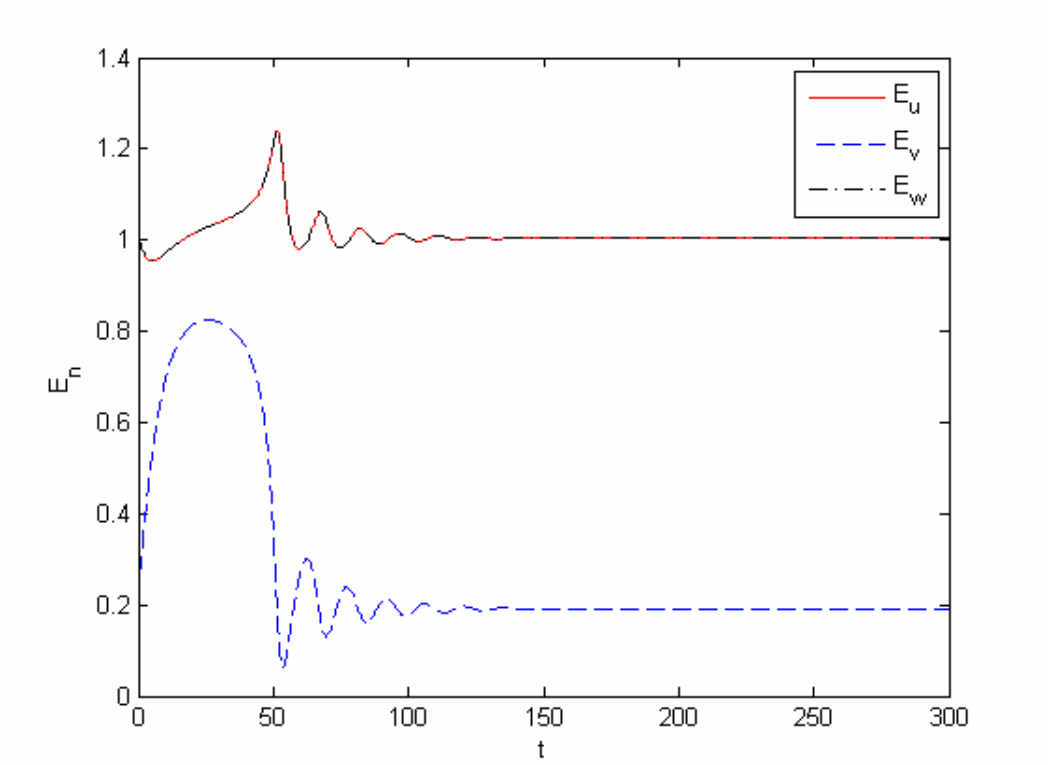

Fig. 5.4 Time evolution of E_n for the ABC flow in the entire domain (with k =1 and R = 11) obtained using PMNIM. The initial condition is one of the secondary solutions which evolve towards the trivial solution initially, however, it returns back to the secondary solution for this value of R.

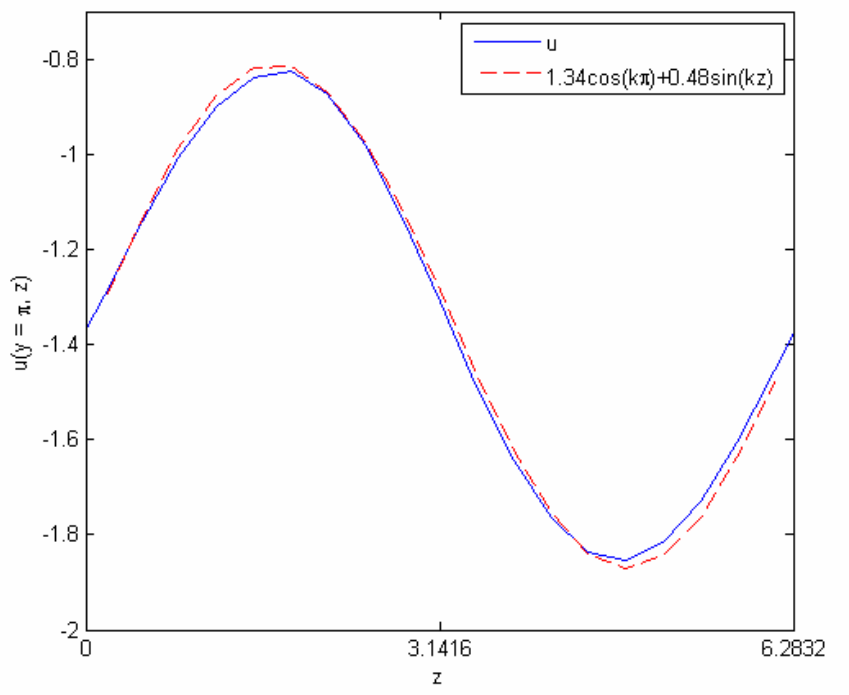

Fig. 5.5 (a) Plot of the u - component of the velocity for $R = 11$ as a function of z. The velocity is plotted at $y = \boldsymbol{\pi}$.

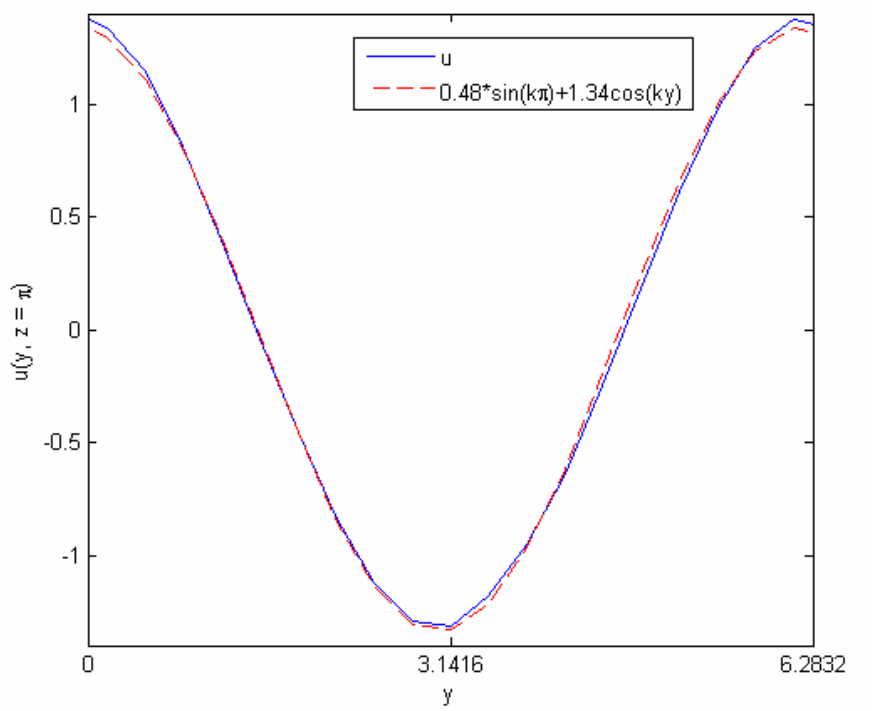

Fig. 5.5 (b) Plot of the u - component of the velocity for $R = 11$ as a function of y. The velocity is plotted at $z = \boldsymbol{\pi}$.

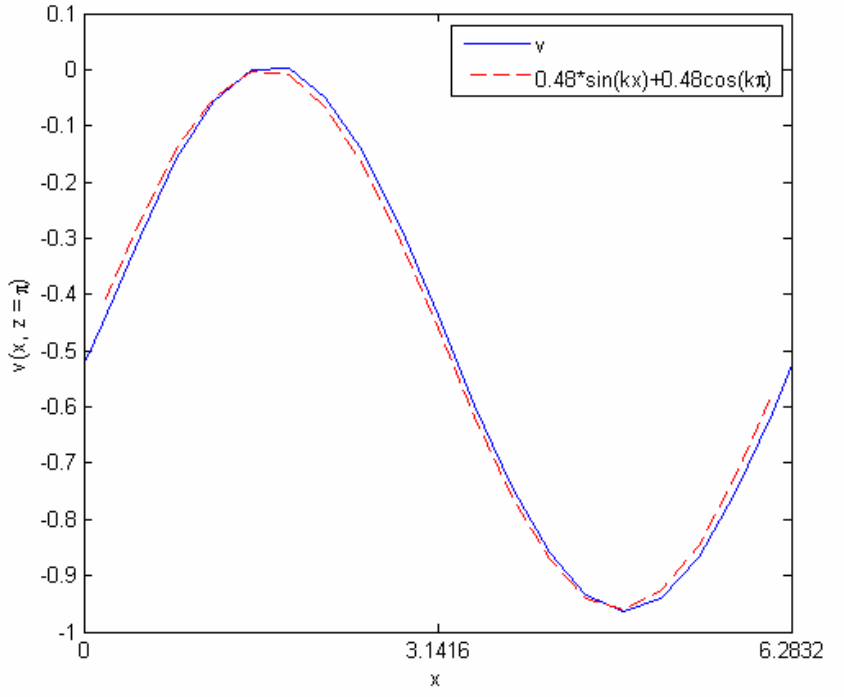

Fig. 5.6 (a) Plot of the *v* - component of the velocity for $R = 11$ as a function of *x*. The velocity is plotted at z = **π.**

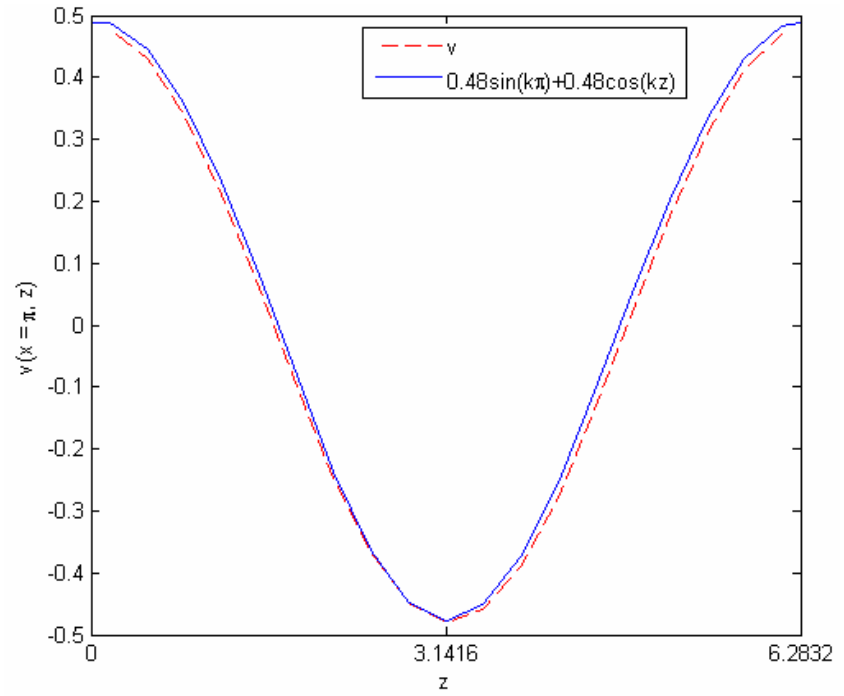

Fig. 5.6 (b) Plot of the *v* - component of the velocity for $R = 10$ as a function of *z*. The velocity is plotted at $x = \boldsymbol{\pi}$.

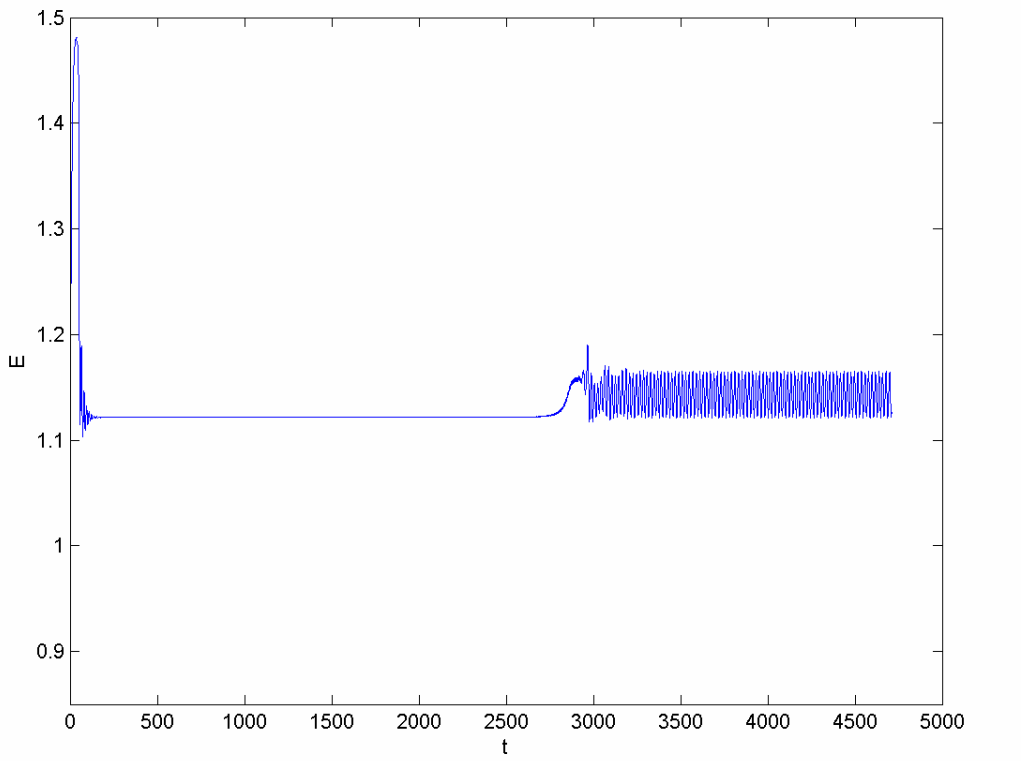

Fig. 5.7 (a) Time evolution of the average energy for the ABC flow in the entire domain, with k =1 for R = 14.9 obtained using PMNIM.

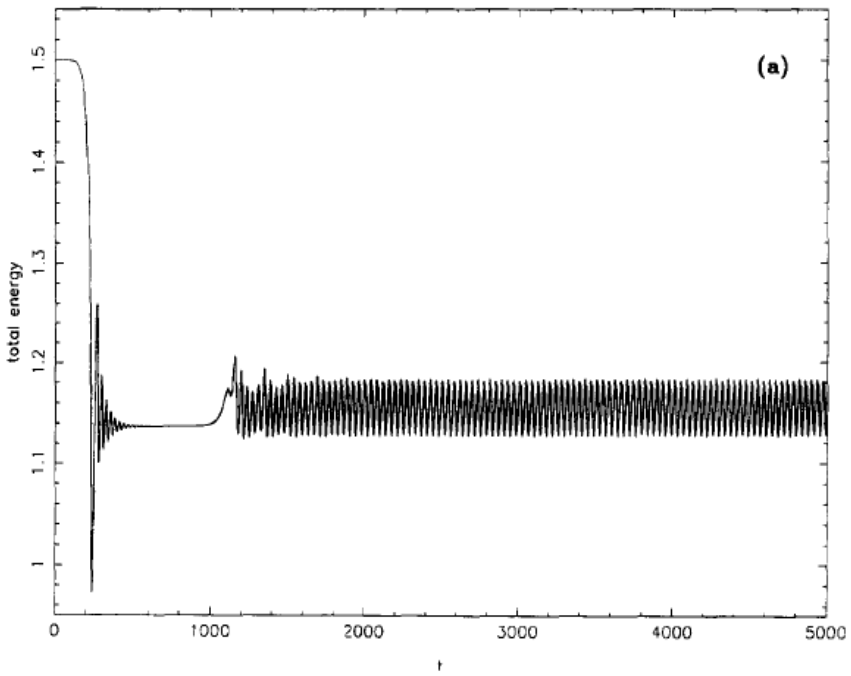

Fig. 5.7 (b) Evolution of the average energy for the ABC flow in the entire domain (with k =1 for R = 14.6). This result is obtained by Podvigina (1994) using the Fourier pseudo spectral method.

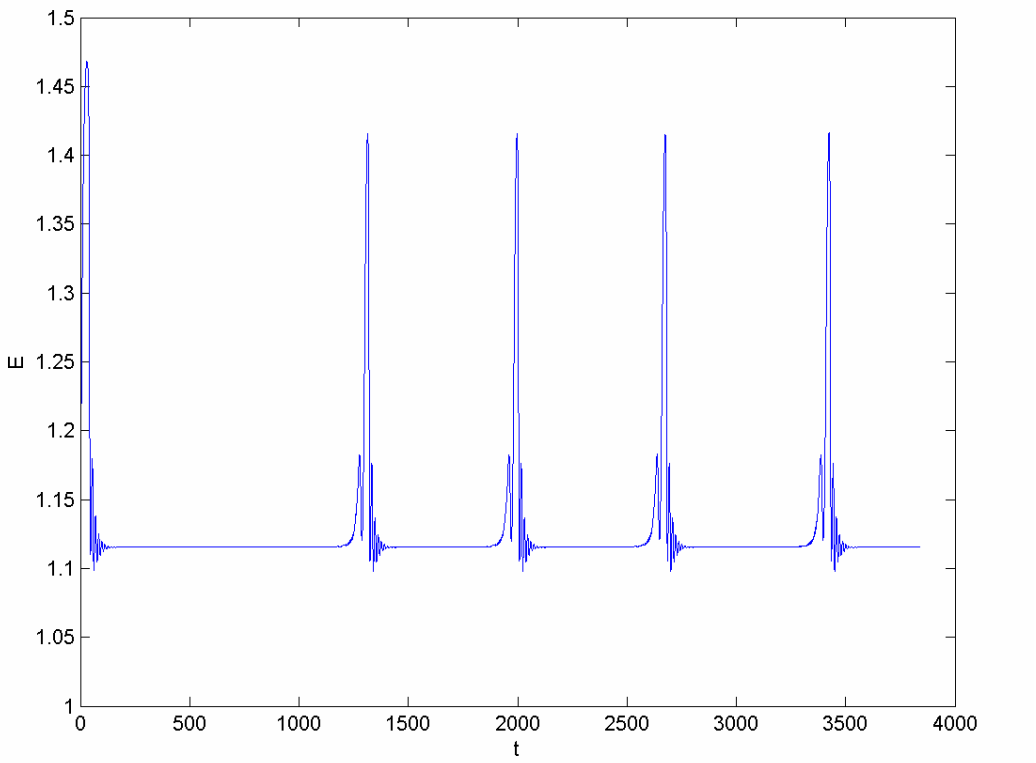

Fig. 5.8 (a) Time evolution of the average energy for the ABC flow in the entire domain (with k =1 for R = 15.5) obtained using PMNIM.

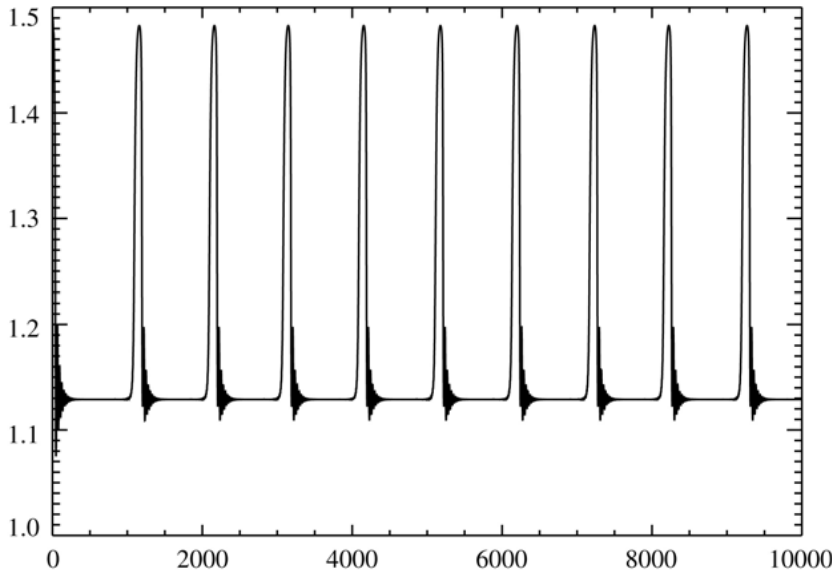

Fig. 5.8 (b) Time evolution of the average energy for the ABC flow in the domain (with k =1 for R = 15.3). This result is obtained by Podvigina (2006) using the Fourier pseudo spectral method.

alternates between the trivial solution and the secondary solution. The solution, however, stays at the secondary solution for much longer period of time.

In the two figures discussed above, it is seen that there is some discrepancy between the results obtained using the PMNIM and the FPSM. This difference is explained by the fact that the perturbations are different in the two schemes. It should also be noted that very small changes in the value of R may result in significant difference in the flow evolution. (All the flow regimes which exist between R = 14.5 to R = 15.5 are not shown here for the sake of brevity.) In fact, there are some discrepancies between the results given by Podvigina (1994) and Podvigina (2006) for the same value of R. Furthermore, it can be seen in Fig. 5.7 that the time for which flow remains in the steady state, before achieving the periodic solution, is different for the PMNIM and FPSM scheme. Similarly, there are some small differences between the solutions shown in Fig. 5.8, obtained by the two different methods. These differences are again the result of different type of perturbations and/or numerical errors. It has been discussed by Lloyd (2005) (in the context of heteroclinic cycles in the one-dimensional complex Ginzburg–Landau equation) that the differences in numerical errors in the solution of these equations can result in changes in the duration of the time the solution spends at a particular fixed point.

The E_n and E, which are obtained at R = 17, using the PMNIM, are plotted in the Fig. 5.9. The results of the FPSM are also shown for the comparison. The plots of the FPSM scheme, however, in this figure and the subsequent figures are not for the E_n. Instead these are for the projections of the Fourier modes of the three velocities. However, the heteroclinic cycles obtained from the two schemes can be compared in

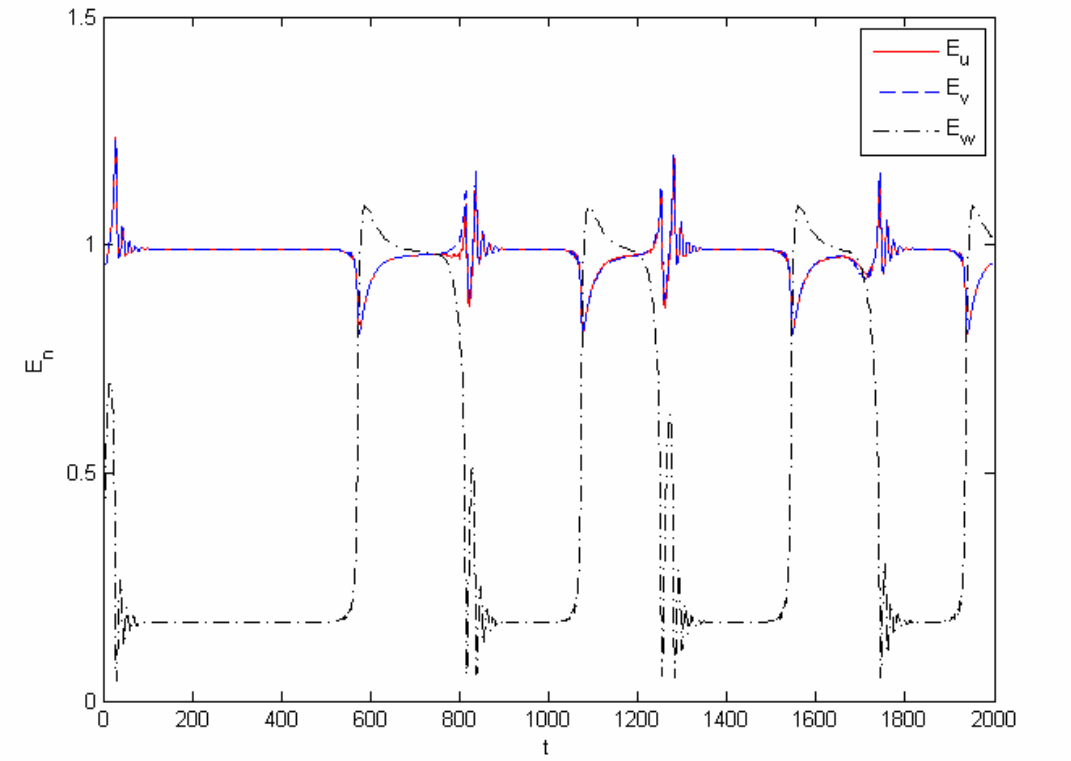

Fig. 5.9 (a) Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 17) obtained using PMNIM.

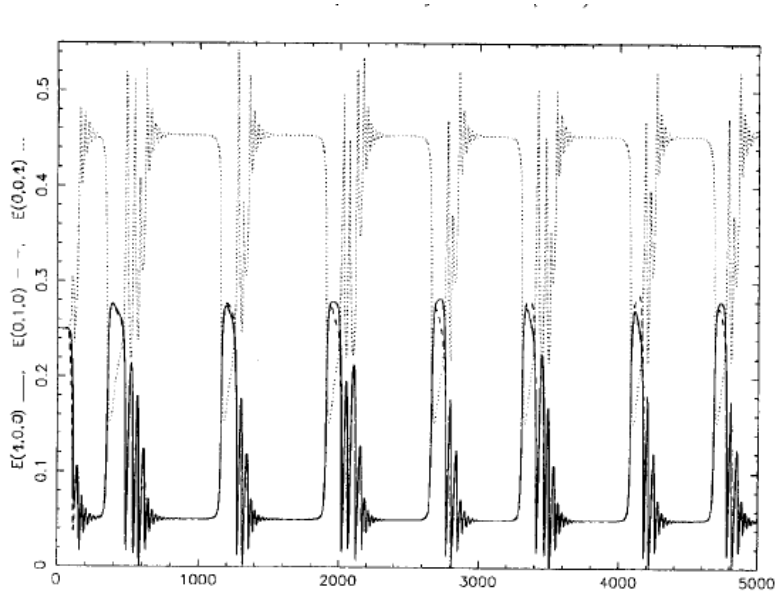

Fig. 5.9 (b) Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 17). This result is obtained by Podvigina (2006) using the Fourier pseudo spectral method. This plot is a projection of the Fourier mode of the velocity. However, it can be seen that the heteroclinic cycles are the same as in panel (a) of this figure.

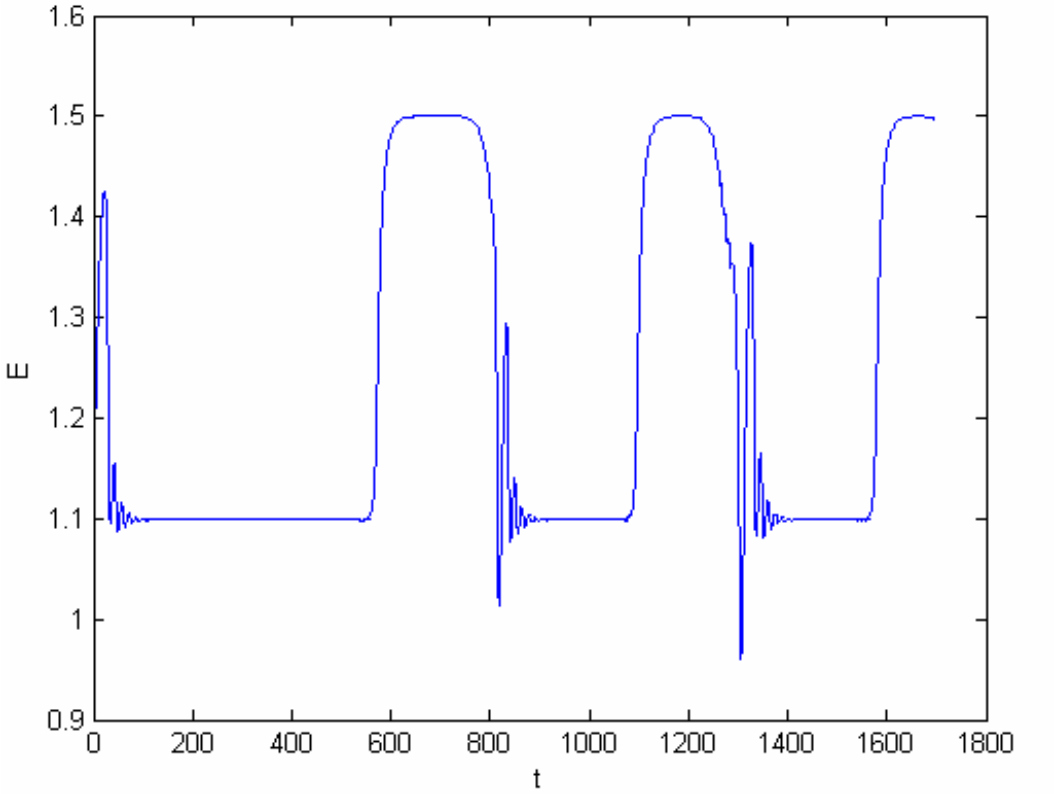

Fig. 5.9 (c) Time evolution of the average energy for the ABC flow in the domain (with k =1 for R = 17) obtained using PMNIM.

quite straightforward manner. For example, from Fig. 5.9, it can be concluded that the solution has a heteroclinic cycle which consists of the sequences of the trivial solution and the secondary solution. Although different quantities are being plotted the conclusion is the same for the plots obtained using the two different schemes. It is also observed that during the heteroclinic cycles, solution returns to the same secondary solution after visiting the trivial solution for short periods of time. It should also be noted that the intervals between these cycles are irregular.

Figure 5.10 shows the results obtained using PMNIM as well as those obtained using the FPSM for $R = 20$. The interesting point to note is the fact that the flow visits all the three permutations of the secondary solution during its evolution in time. It also briefly visits the trivial solution when it interchanges from one secondary solution to the other secondary solution. The plots, similar to those in the previous figure, are shown in Figs. 5.11 and 5.12 for $R = 23$ and $R = 25$, respectively. However, FPSM plots are not shown at $R = 23$, as these are not available. At $R = 23$, it is observed that the heteroclinic cycles are similar to those seen at $R = 20$. However, the transition from one steady state to another is quite chaotic. The plots at $R = 25$ show that the flow has become fully chaotic and that the flow consists of heteroclinic cycles of the chaotic invariant sets.

It is noted that the solutions obtained for higher values of R using the PMNIM, agree very well with those obtained using the FPSM, unlike the solution at lower values of R which show small differences between the two results. This is explained by the fact that the perturbations required to destabilize the flow for the lower values of R are quite large, and therefore different schemes having different type of numerical errors can result in slightly different behaviors. At larger values of R, the perturbations required for the

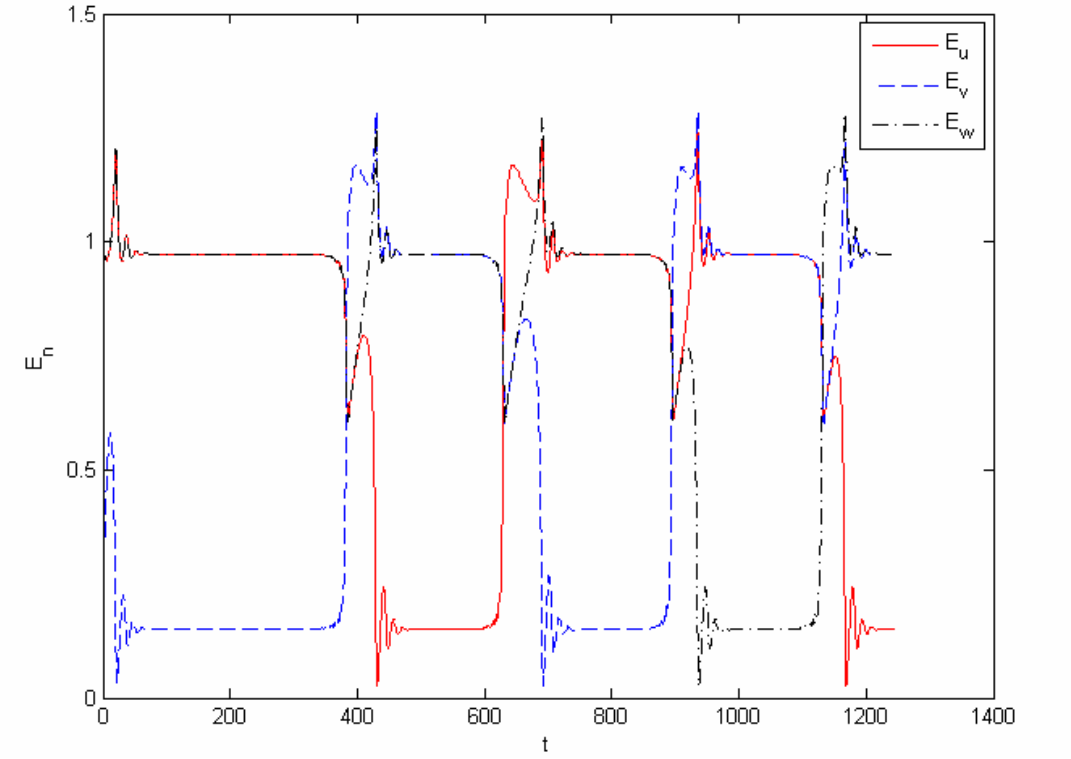

Fig. 5.10 (a) Time evolution of E_n for the ABC flow in the domain (with k =1 for R = 20) obtained using PMNIM.

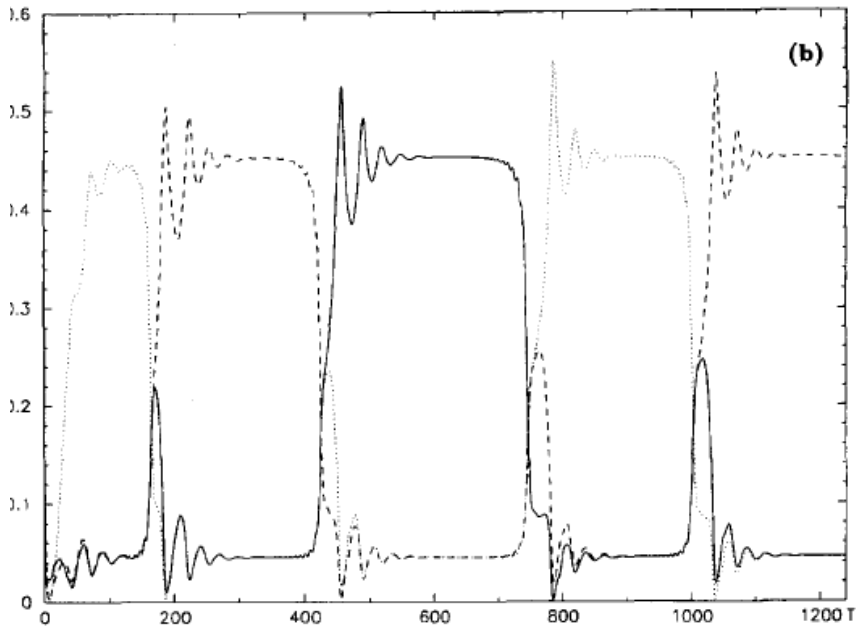

Fig. 5.10 (b) Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 17). This result is obtained by Podvigina (2006) using the Fourier pseudo spectral method. This plot is a projection of the Fourier mode of the velocity. However, it can be seen that the heteroclinic cycles are the same as in panel (a) of this figure.

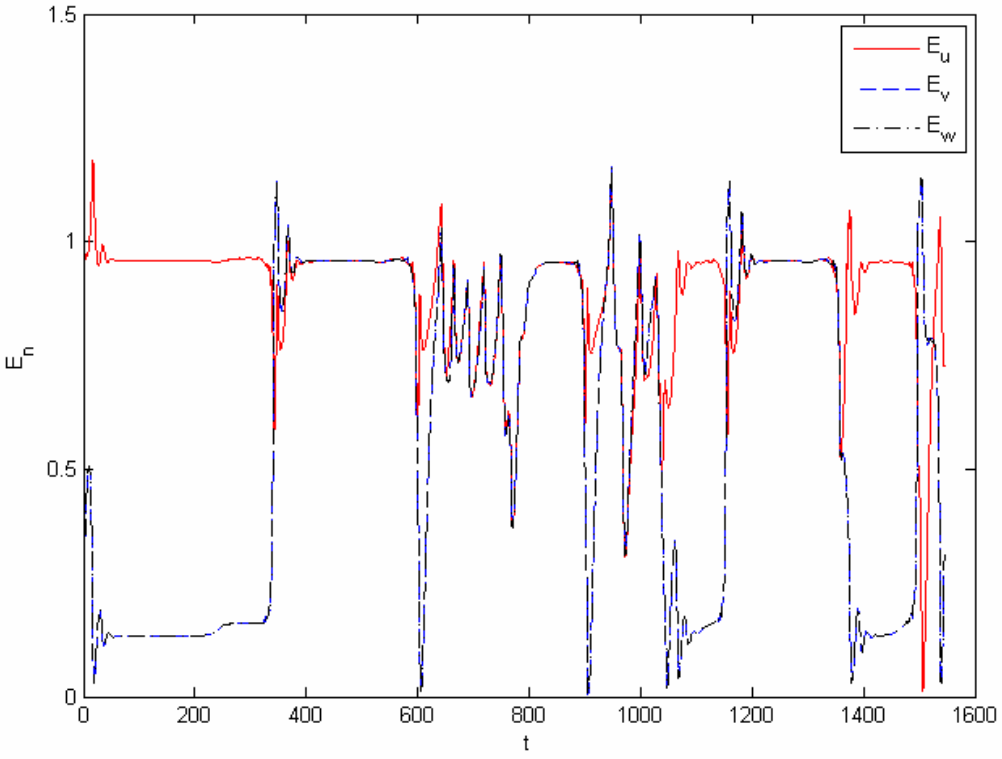

Fig. 5.11 Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 23) obtained using PMNIM.

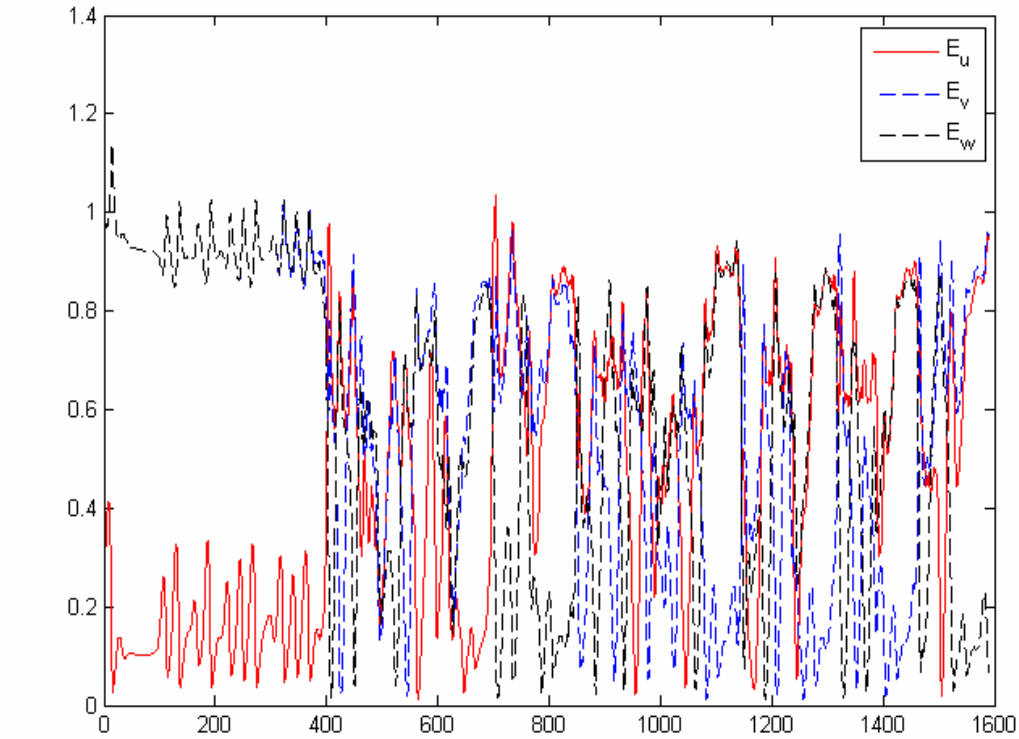

Fig. 5.12 (a) Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 25) obtained using PMNIM.

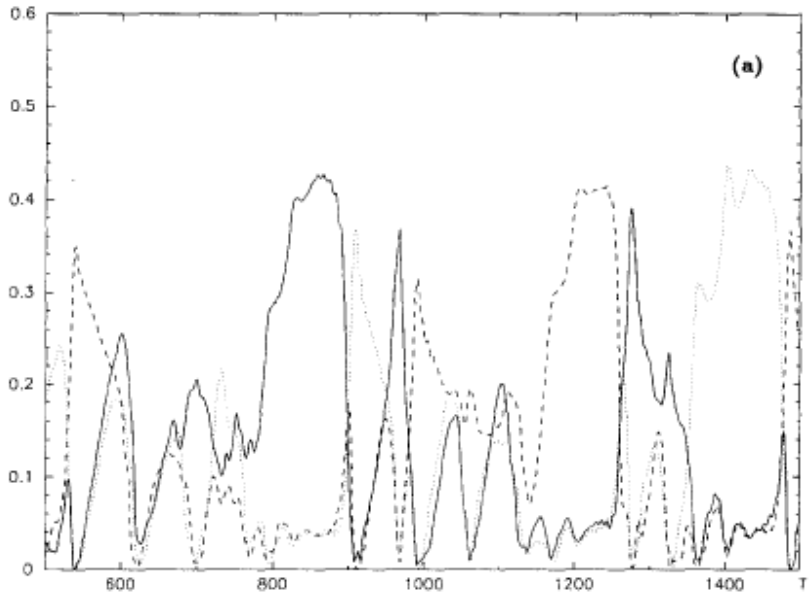

Fig. 5.12 (b) Time evolution of E_n for the ABC flow in the entire domain (with k =1 for R = 25). This result is obtained by Podvigina (2006) using the Fourier pseudo spectral method. This plot is a projection of the Fourier mode of the velocity. However, it can be seen that the heteroclinic cycles are the same as in panel (a) of this figure.

instability are quite small; and therefore, the (small) difference in the size (and type) of the perturbations does not change the solution. In fact, at very large R flow becomes linearly unstable and infinitesimally small perturbation will cause the instability. Hence, the size of the perturbation for these values of R becomes immaterial.

5.5 Results for $k = 2$ case

The computations for the $k = 2$ case have been carried out using a 28^3 grid and timesteps of 0.01. The above parameters are kept the same for all the values of R. The trivial solution, with no explicit perturbations, is used as the initial condition for all the simulations. Some calculations are carried out with grid of 40^3 in order to study the grid size effects. It was observed that except for the lengthening of some of the heteroclinic cycles the differences between the results obtained by using the two grid sizes are insignificant. This lengthening of the cycles, as discussed before, is because of the different numerical errors in the solution and has been reported in the past (Lloyd 2005).

Figure 5.13, that corresponds to Fig. 5.1 for the $k = 1$ case, shows the validity of the trivial solution at low values of R. The results in this figure are obtained for $R = 4$. Figure 5.14 shows the plot of the E_n for $R = 5$. It can be observed from this figure that for $R = 5$, the solution, starting from the trivial solution, evolves into a secondary steady state solution. Obviously, as in the case of $k = 1$, the permutations of this solution will also be the solutions, due to the symmetry in the governing equations of the flow. This solution loses its stability after evolving for a long time and oscillatory solution can be seen. The interesting fact to note is that, unlike the $k = 1$ case, E_v and E_w also separate into two different branches. Another difference from the $k = 1$ case is that, as can be seen

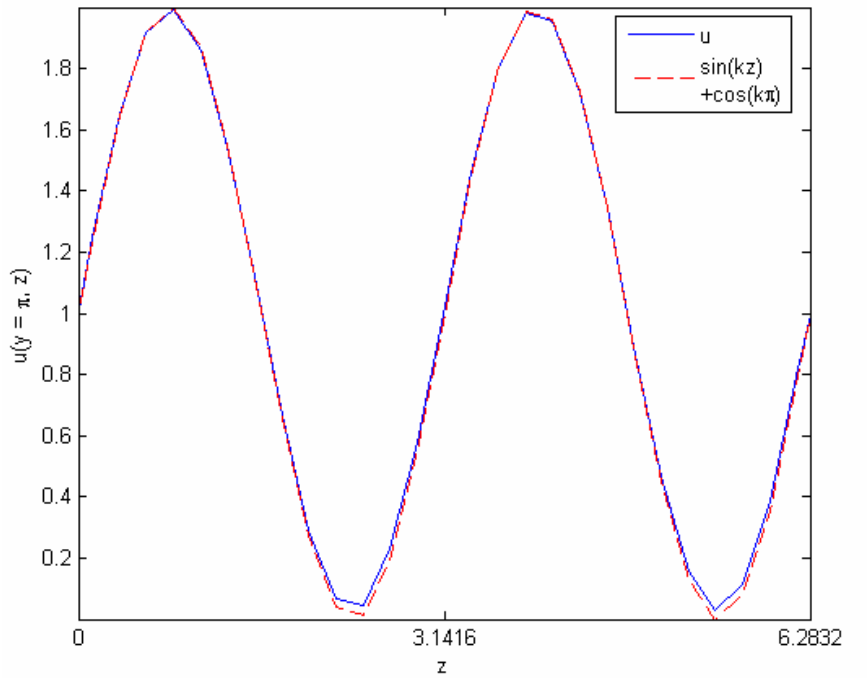

Fig. 5.13 (a) Plot of the u - component of the velocity for $R = 4$ (with $k = 2$) as a function of z. The velocity is plotted at $y = \boldsymbol{\pi}$.

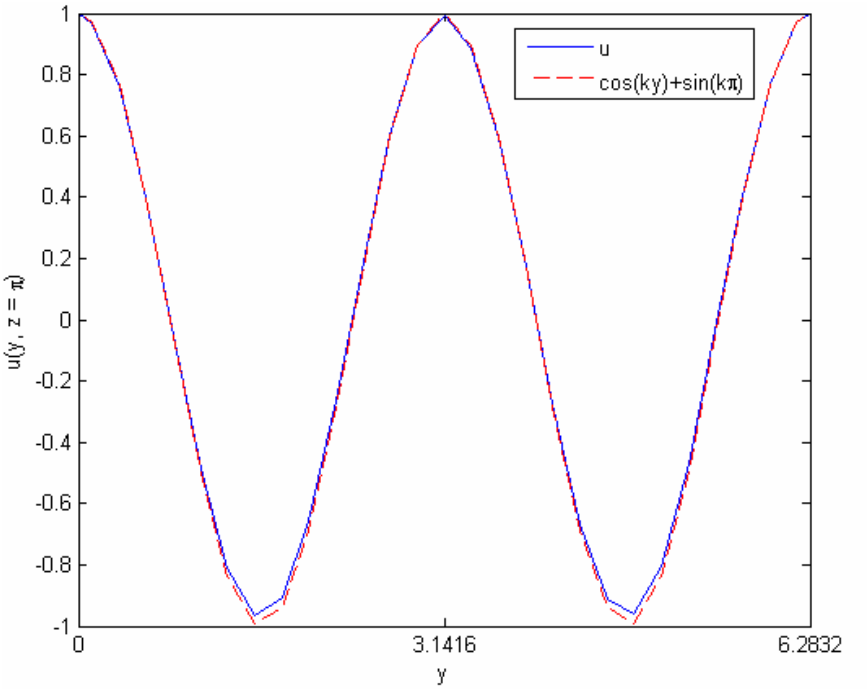

Fig. 5.13 (b) Plot of the u - component of the velocity for $R = 4$ (with $k = 2$) as a function of z. The velocity is plotted at $z = \boldsymbol{\pi}$.

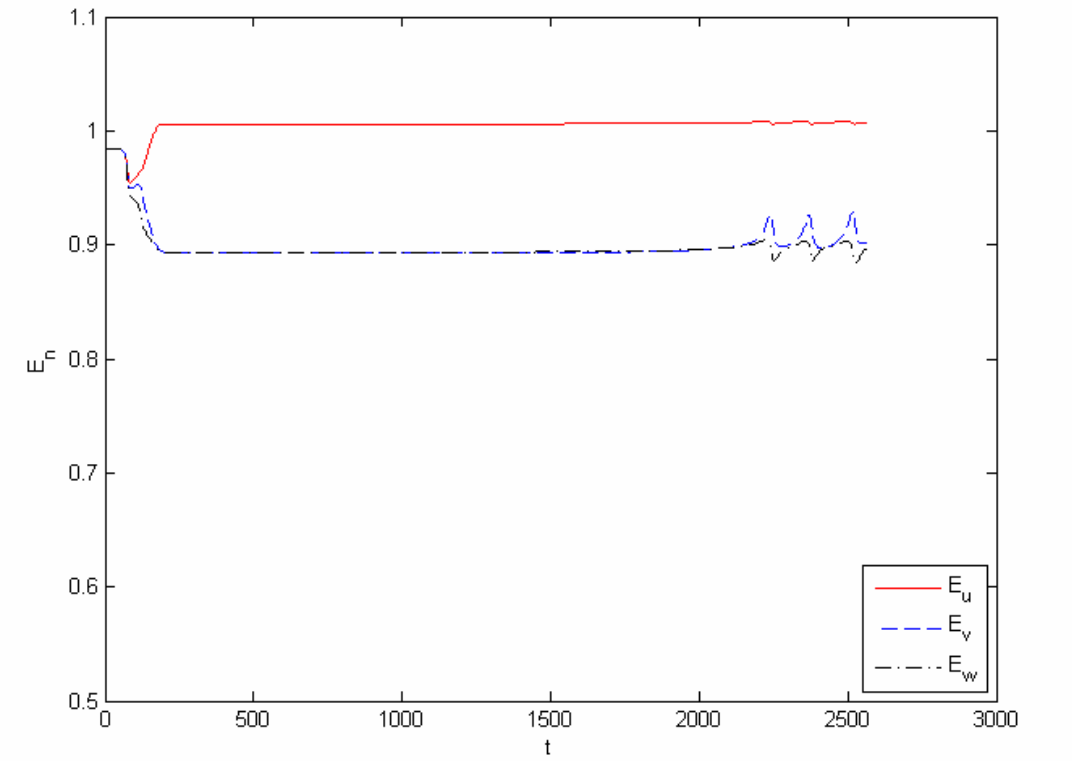

Fig.5.14 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 5$) obtained using PMNIM.

from this and subsequent figures, the solutions for the $k = 2$ case are strongly dependent on the value of R, even at the low values of R.

Figure 5.15 shows the plots of the secondary steady state solution (i.e. the plots are obtained at a time which is before the onset of the instability). In Fig. 5.15 (a), u-component of the velocity is plotted as a function of z at $x = y = \pi$. The panel (b) of the same figure shows the plot of the u-component as a function of y at $x = z = \pi$. It is interesting to note that the u-component of the velocity at $x = \pi$ and $y = \pi$ (as seen from Fig. 5.15(a)), not only consists of the sinusoidal function with wavenumber $k = 2$ but also the wavenumber $k = 1$. It is also unlike the secondary solution obtained for the $k = 1$ case, in which the solution consists of only one wavenumber. The u- component of the velocity at $x = y = \pi$ is well captured by the following,

$$u(x=\pi, y=\pi, z) = c_{1s}\sin(z) + c_{2s}\sin(2z) + c_{2c}\cos(2z) + c_0 . \tag{5.15}$$

where, the Fourier coefficients (c_{1s} etc.) are shown in Fig. 5.14 (a). These Fourier coefficients for the sine terms (c_{ns}) in the above equation are found as follows,

$$c_{ns} = \frac{\sum_{k=1}^{m} u_k(x=\pi, y=\pi, z_k)\sin(nz_k)}{\sum_{k=1}^{m}\sin(nz_k)\sin(nz_k)} . \tag{5.16}$$

where u_k is a vector consisting of the u- velocity at the discrete points z_k (which is z coordinate of the center of a cell) obtained using the numerical scheme and m is the number of the cells in the z-direction. The coefficient for the cosine terms (c_{nc}) can be found be replacing $\sin(nz_k)$ by $\cos(nz_k)$ in the above equation. The constant term (c_0) is the average value of the u_k.

Similar observation, such as presence of the wavenumber $k = 1$, can be made from Fig. 5.15 (b) about the u- component of the velocity at $x = z = \pi$. The coefficients for the various terms are shown in the figure. Similar observations are made about the solution for the other velocity components as well (not shown here).

In the following figures, E_n is plotted for the various values of R. In Fig. 5.16, plots are shown for $R = 6$. From these plots it can be seen that the oscillations in the E_n have become larger than those for $R = 5$. There is large separation between E_v and E_w after the flow has evolved over a long time. It is also seen that the period of the oscillations becomes shorter as time increases.

Although, the solution at $R = 8$ (Fig. 5.17) is similar to the previous solutions, it shows the heteroclinic cycles which alternate between the different permutations of the set of the secondary solutions. It should also be noted that these cycles consist of the set of oscillatory solutions. In addition, it is also pointed out that the oscillatory solution returns to the steady state solution after a few heteroclinic cycles. This in itself should be considered a heteroclinic cycle consisting of a sequence of oscillatory and the steady state solutions. Therefore, it can be concluded that the two types of the heteroclinic cycles exist simultaneously or "overlap" at this value of R.

Figure 5.18 shows the plots of E_n at $R = 10$. The Hopf bifurcation, resulting in a decaying periodic solution with constant time periods, can be seen in these plots. The evolution of the flow is shown in these plots only for a short period of time as further evolution of this flow is similar to that for $R = 8$. Note that these oscillations are quite different from the oscillations seen in the previous figures. Panel (b) in this figure is the

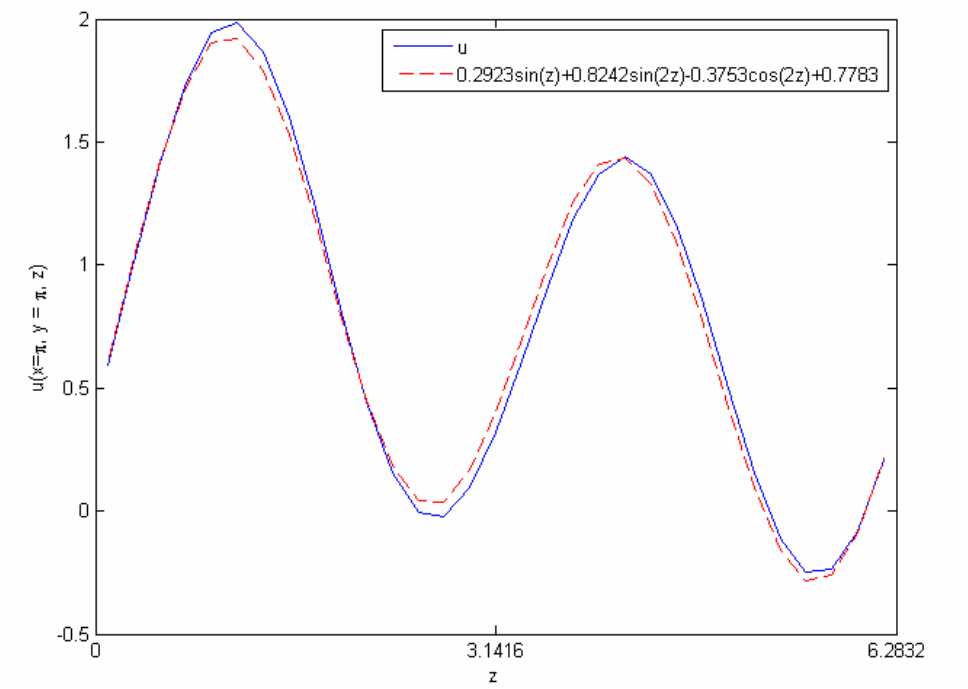

Fig. 5.15 (a) Plot of the u - component of the velocity for $R = 6$ (with $k = 2$) as a function of z. The velocity is plotted at $x = y = \boldsymbol{\pi}$.

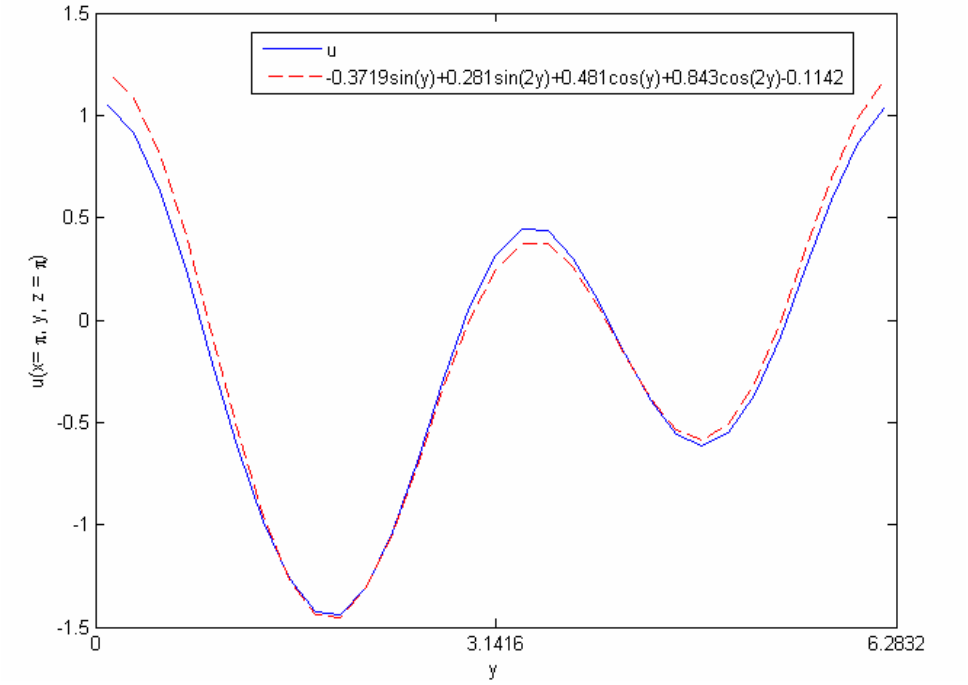

Fig. 5.15 (b) Plot of the u - component of the velocity for $R = 6$ (with $k = 2$) as a function of y. The velocity is plotted at $x = z = \boldsymbol{\pi}$.

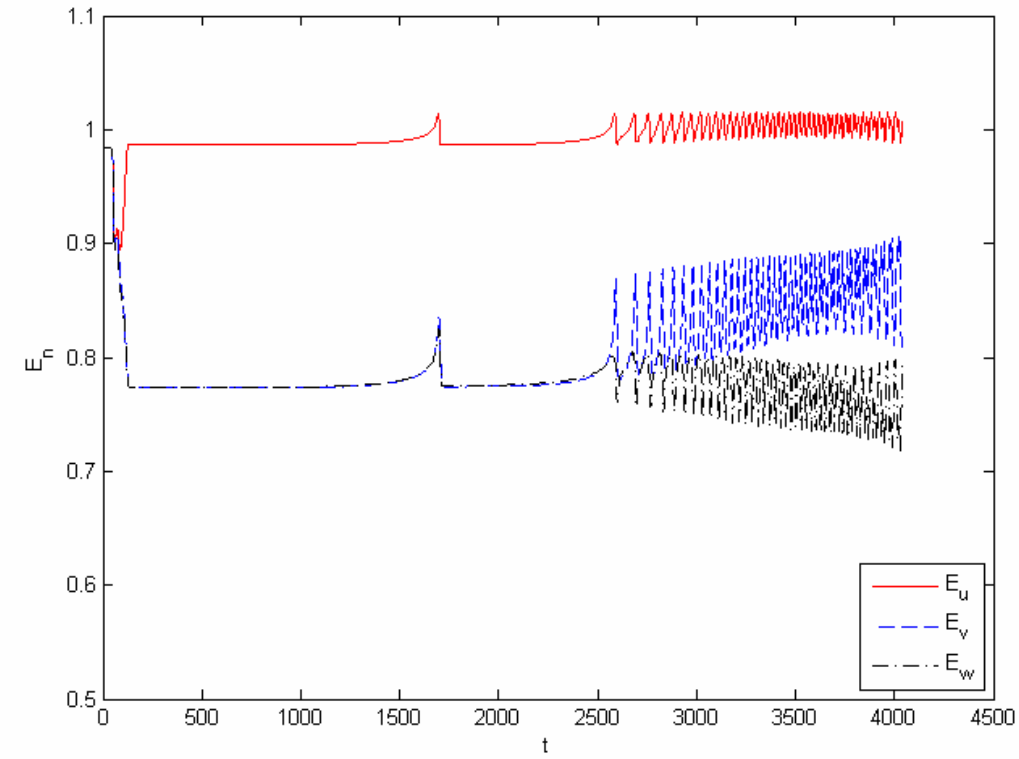

Fig. 5.16 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 6$) obtained using PMNIM.

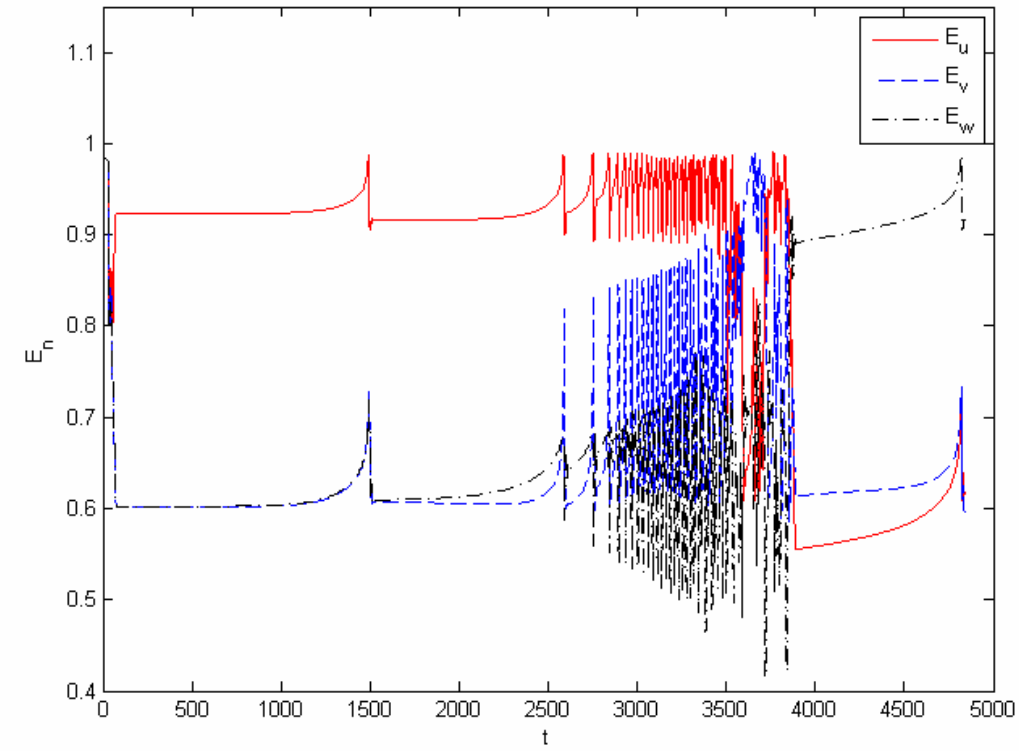

Fig. 5.17 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 8$) obtained using PMNIM.

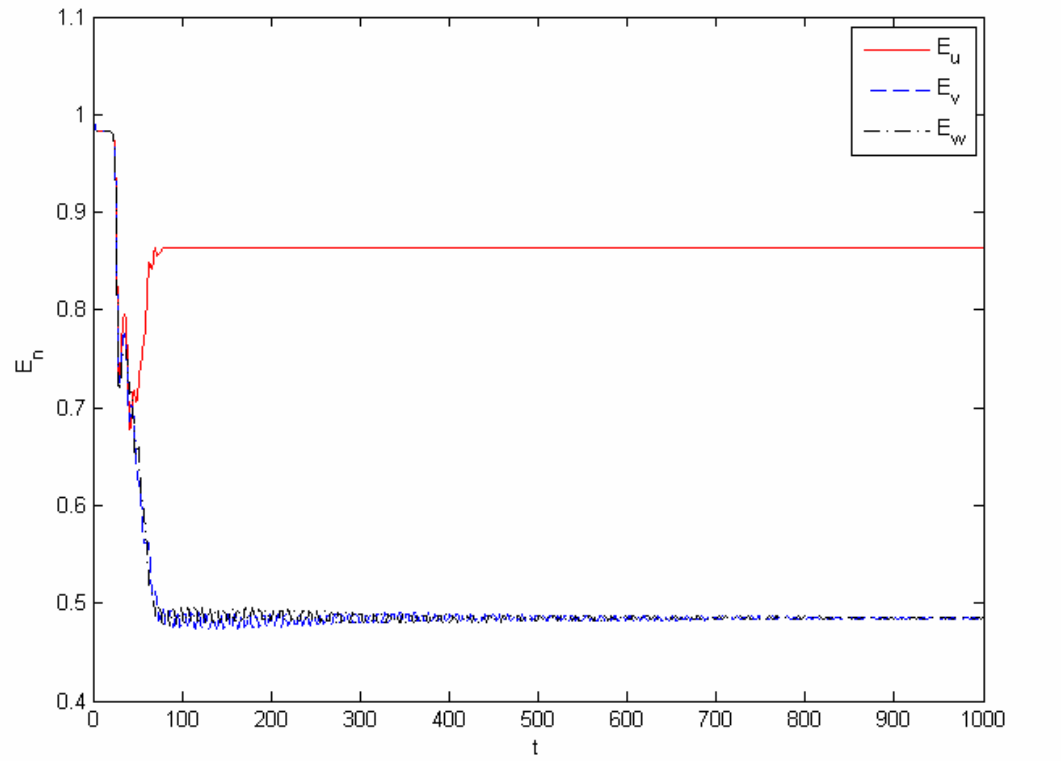

Fig. 5.18 (a) Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 10$) obtained using PMNIM.

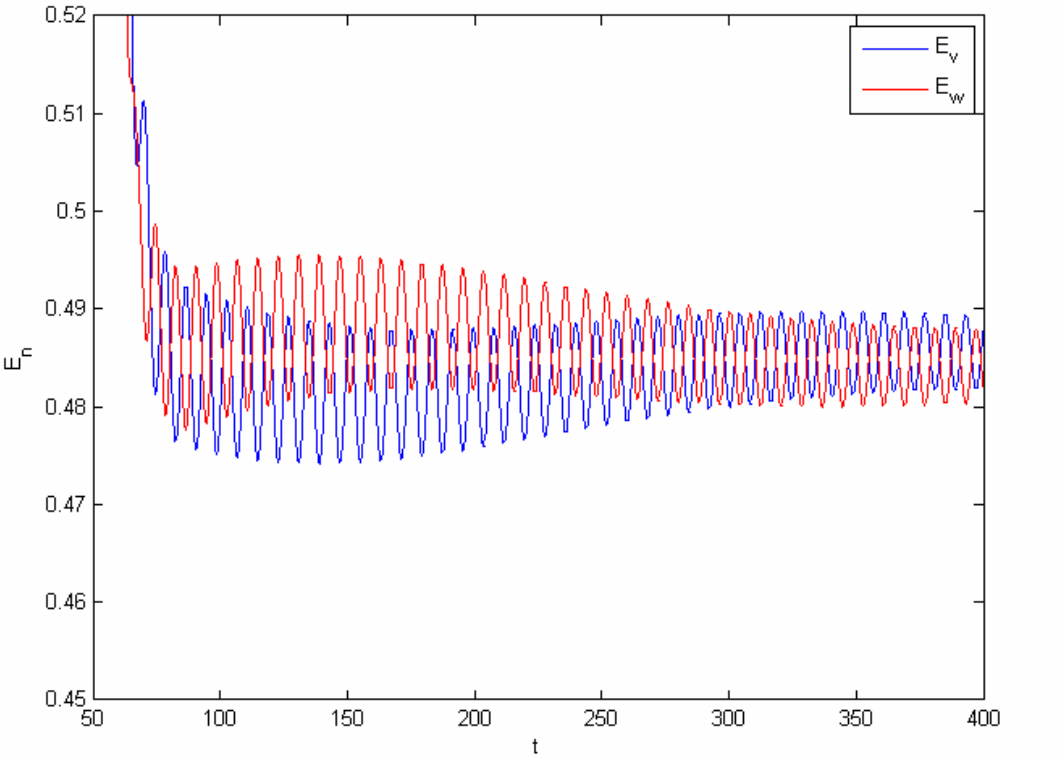

Fig. 5.18 (b) Time evolution of E_v and E_w for the ABC flow in the entire domain (with $k = 2$ for $R = 10$) obtained using PMNIM. This figure is zoomed version of the panel (a).

zoomed in version of the E_v and E_w plots showing opposite phase of these two quantities.

At $R = 11$ and $R = 12$, overlapping or simultaneous existence of the two types of the heteroclinic cycles can be seen again. The first of the heteroclinic cycle is manifested in the cycling between the two permutations of the set of the secondary solutions, which is observed in the interchange of E_v and E_w in Figs. 5.19 and 5.20. The second heteroclinic cycle can be seen as sequences of periodic and steady state solutions. These Hopf-steady state heteroclinic cycles have been previously reported in the literature (e.g. Melbourne 1989). The difference between the solutions at $R = 11$ and $R = 12$ is that the flow at higher R has become somewhat chaotic.

The flow at $R = 13$ (Fig. 5.21) is considerably more chaotic than the flows at relatively lower R. However, there are durations in which flow can be considered having a periodic solution similar to that observed in the previous figure. Further increase in the value of R results in the flow becoming fully chaotic. The plots of E_n for $R = 14$ and $R = 16$ are shown in Figs. 5.22 and 5.23, respectively. It is observed that in both the cases the flow consists of heteroclinic cycles of the chaotic invariant sets. However, duration between the cycles is smaller for the flow at higher values of R.

It should be pointed out here that attempt has not been made to determine the critical values of R for the onset of different phenomena observed in these simulations. This is partly due to the fact that the type of flow field that results from these simulations is sensitive to the type of perturbations introduced and hence, the type and size of numerical errors can affect the threshold or critical values of R. This is, in fact, true for

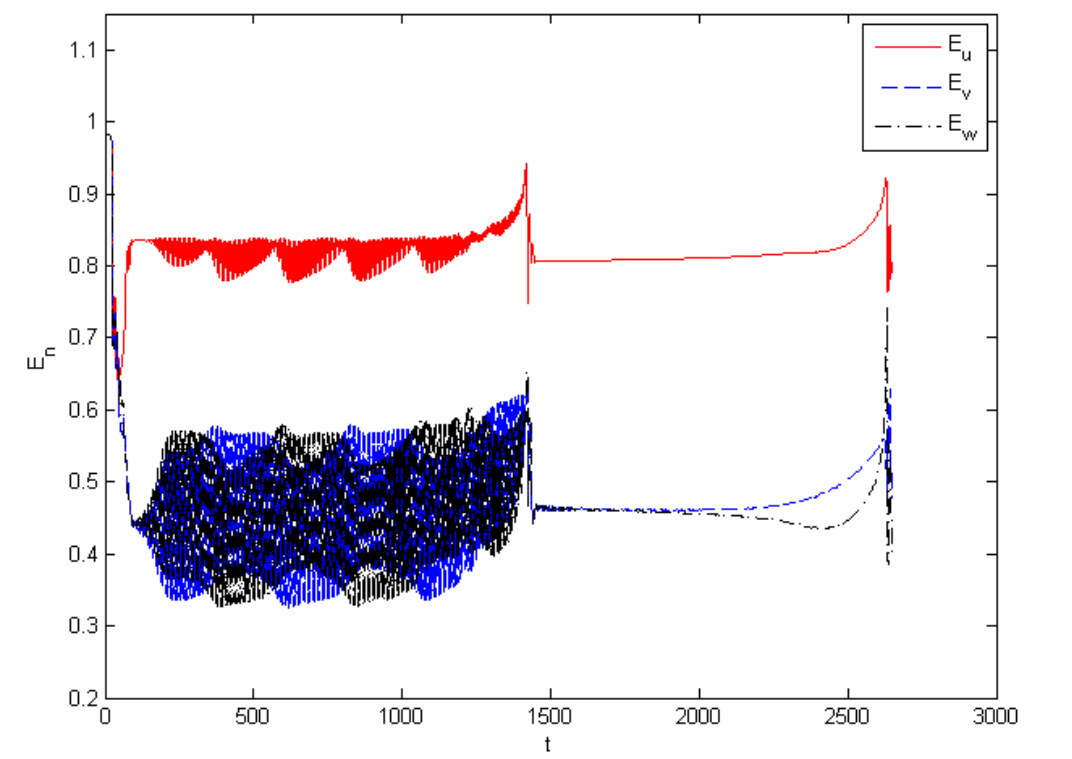

Fig. 5.19 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 11$) obtained using PMNIM.

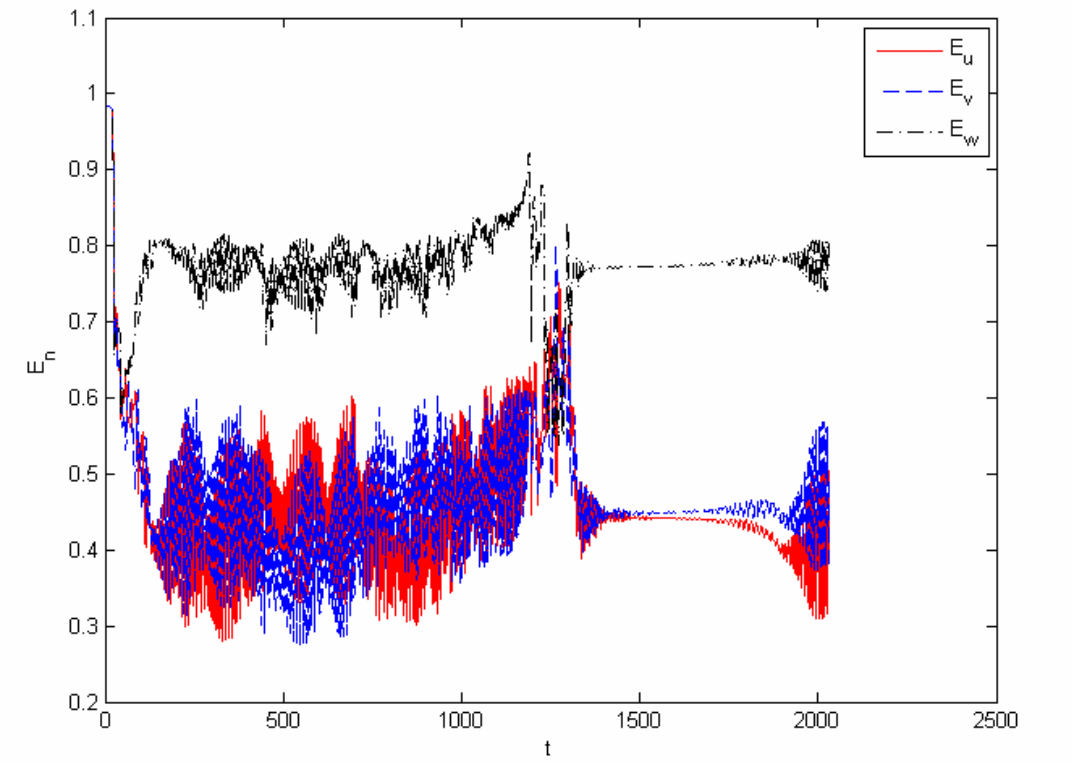

Fig. 5.20 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 12$) obtained using PMNIM.

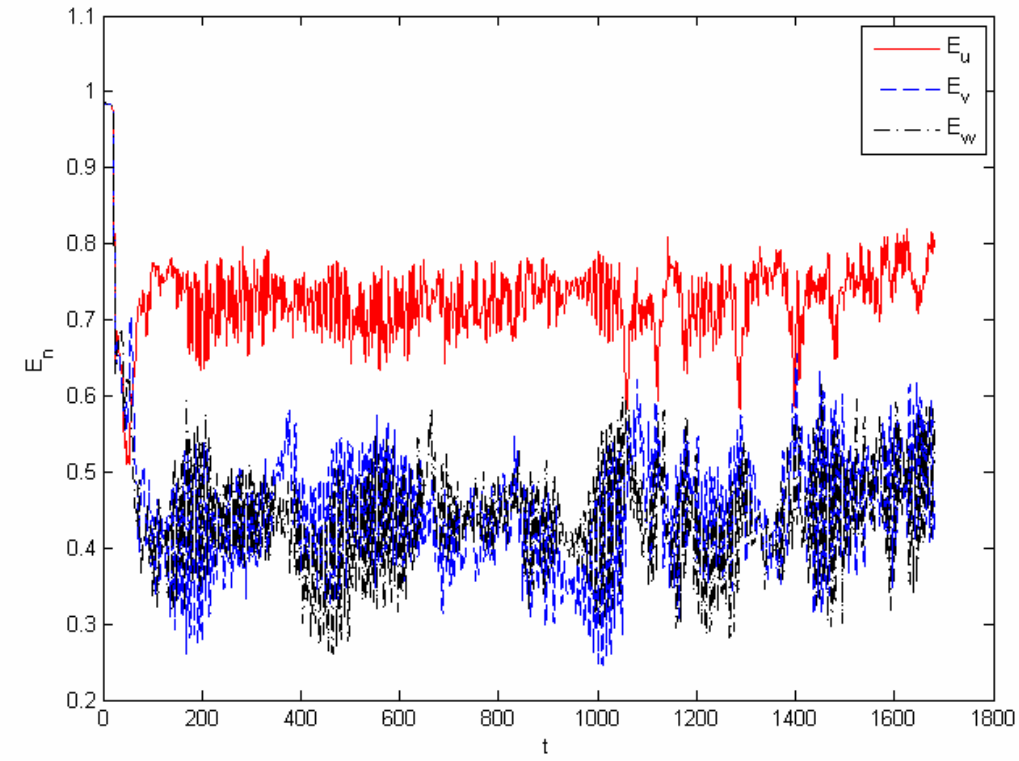

Fig. 5.21 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 13$) obtained using PMNIM.

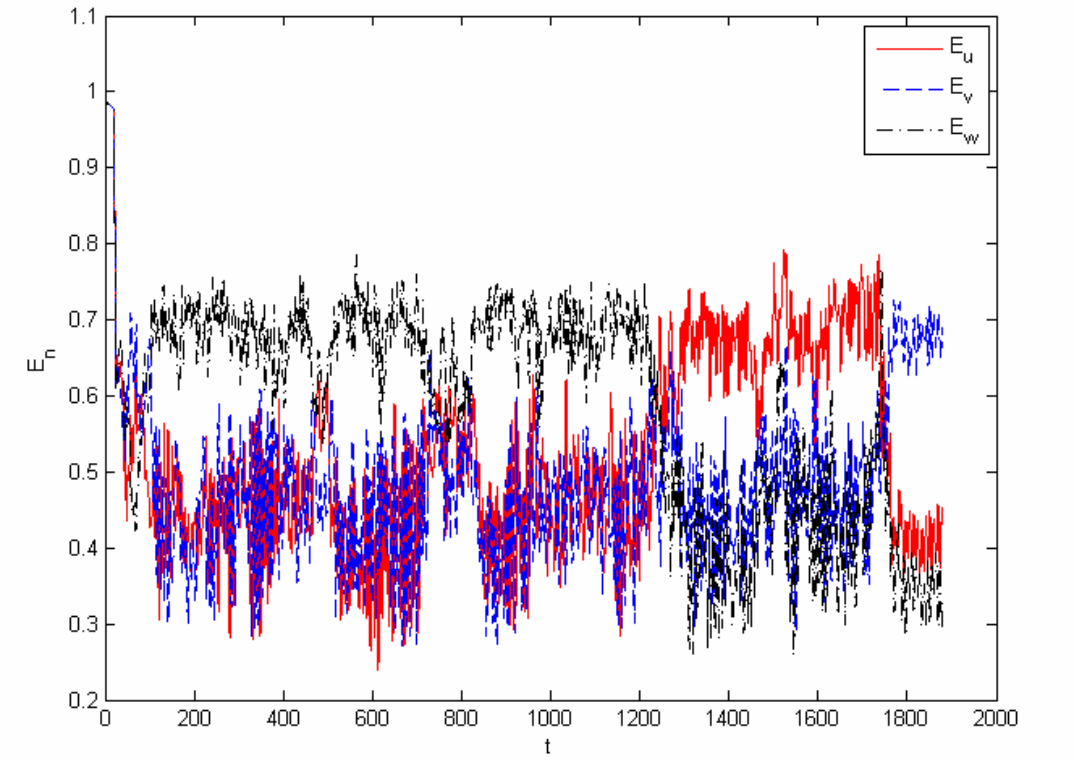

Fig. 5.22 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 14$) obtained using PMNIM.

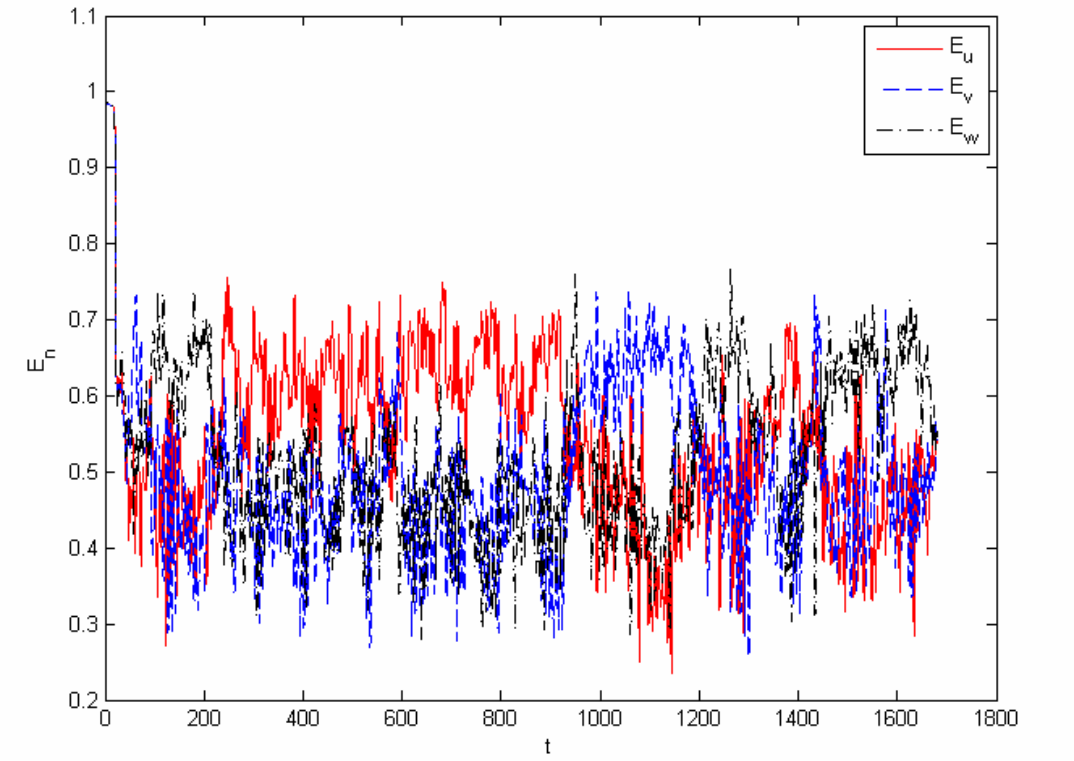

Fig. 5.23 Time evolution of E_n for the ABC flow in the entire domain (with $k = 2$ for $R = 16$) obtained using PMNIM.

such simulations with any numerical scheme. However, overall characteristics of the flow are expected to be independent of these perturbations.

5.6 Summary of the DNS using PMNIM

The simulations for the 1:1:1 ABC flow are carried out by using PMNIM for wavenumbers $k = 1$ and $k = 2$. The results for the $k = 1$ case are available in the literature and have been obtained in the past using Fourier pseudo spectral method (Podvigina 1994, Podvigina 2006). The results, obtained using the PMNIM, for the case of $k = 1$, are compared with the results reported in the abovementioned works. The grid sizes and timesteps used in this dissertation are similar to those used by the Podvigina (1994). It is concluded that the results obtained with the PMNIM match quite well with those previously obtained results using FPSM, especially at higher values of R. The small discrepancies observed at the lower values of R are presumed to be due to different perturbations for these two different schemes.

The simulation for the case of $k = 2$ yield a set of secondary solutions as was the case of $k = 1$. However, unlike the case of $k = 1$, these solutions consist of two wavenumbers $k = 1$ and $k = 2$. These secondary solutions are in addition to the trivial solution which is the only solution for small values of R. The secondary solutions become unstable at $R = 5$. Note that the R at which this flow becomes unstable is much lower than $R = 13$ at which the flow for $k = 1$ becomes unstable. This is expected as smaller length scales are being enforced in the flow with $k = 2$. At higher values of R (approximately between 7 and 12), there are heteroclinic cycles which visit different permutations of the secondary set of solutions during the evolution of the flow in time. In addition, there are

heteroclinic cycles which are sequences of the steady state and oscillatory solutions. It is interesting to note that the abovementioned two types of heteroclinic cycles result in quite complex evolution of the flow for the case of $k = 2$. The flow becomes chaotic as R is further increased and consists of heteroclinic cycles of the chaotic invariant sets for values of $R > 14$.

Chapter 6

Summary, Conclusions and Future Work

6.1 Summary and Conclusions

The use of the NIMs for solving neutron diffusion and transport equations is widespread (Lawrence 1986). Recently, there have been considerable efforts to extend the use of the NIMs to other fields of science and engineering (Horak 1985, Azmy 1983, Elnawawy 1990, Wescott 2001). One of the recent advancements in the application of the NIMs is the development of a modified NIM (MNIM) to solve laminar fluid flow equations (Wang 2003, Wang 2005). In the present work, MNIM is extended to turbulent flows.

In the recent decades, several models have been developed for simulation of turbulent flows. Considering the widespread use of the k-ε modeling approach for flow simulations, a 2D MNIM is developed for the turbulent flow simulations using this model. Several variations of these models are used in the engineering applications. In this dissertation, two of these variations, one for low Reynolds number and one for high Reynolds number, are implemented. The low Reynolds number model implemented is the AKN model (Abe 1990) and the wall function approach for standard k-ε is used for the high Reynolds number model.

The MNIM previously developed for the laminar flows is extended to accommodate the variable eddy viscosity encountered in the k-ε model. The numerical scheme for the k and ε using the NIM approach is also developed. In the development of the scheme, the k and ε equations are split into the convection-diffusion terms and all the other terms in these equations (terms which cannot be included in the convection and diffusion terms) are considered as source terms. The numerical scheme for these equations coupled with the above-mentioned modified MNIM completes the development of the scheme for the models. The appropriate modifications, depending on the particular variation of the k-ε model, are made in the developed numerical schemes for the near wall regions.

The implementation of the other variations is not likely to differ significantly, from the approach used in this dissertation, as underlying features of these variations are quite similar. Moreover, the model equations are largely same away from the walls (except for some small variations in the model constants).

The developed numerical scheme for k-ε model is used to simulate some simple flows. The results obtained from these simulations are compared with the DNS results available in the literature. The comparisons show that results of the simulations in the present work match reasonably well with the DNS results. The discrepancies between the two results are due to limitations of the k-ε model rather than due to the deficiencies in the numerical scheme. The above observation is based on the fact that, in general, the discrepancies seen in the present work are same as those reported for the simulations carried out using other numerical schemes.

In high Reynolds number model, it should be noted that the interface of the two adjacent cells next to the wall in the wall normal direction should lie in the log-layer (Durbin 2001). The grid refinement near the wall is governed by this fact and not on the numerical scheme used for the flow simulations. Similarly, in the low Reynolds number model the grid refinement near the wall depends on the fact the first cell should be inside the viscous layer. Therefore, in both the approaches it is observed that grid refinement near the wall is independent of the accuracy of the numerical scheme used. However, it can be seen that away from the walls the good results are obtained with quite coarse grids.

Although, direct numerical simulation is limited to relatively simple flows and also to relatively low Reynolds numbers, the DNS using MNIM is also explored because it is useful as a research tool. Since, DNS is computationally intensive; therefore, a parallel version of the MNIM (PMNIM) for three dimensional incompressible Navier-Stokes and energy equations is developed to enhance its capability. The dependencies of the variables with each other in the MNIM are complex and unique. Various dependencies between the variables are identified and a communication strategy is developed based on the dependencies of these variables. The domain decomposition paradigm is used for the parallelization of the scheme. The computer code for the PMNIM is developed using MPI (Gropp 1999a). A strategy is also developed to gather all the variables into a single processor after the completion of the simulation, for the post processing purpose. The reliability and efficiency of the parallelization is tested by applying the developed computer code to solve the problem of natural convection in a cubic cavity.

The speedup and efficiency are calculated in two different ways. The first approach is based on the Gustafson's law (Gustafson 1988), in which computational load per processor is kept constant. In this approach, the overall problem size increases with the number of processors. In the other approach (based on Amdahl's law (Amdahl 1967)) the overall problem size is kept fixed. Therefore, problem size per processor decreases as number of processors is increased. Obviously, method of calculating efficiency based on the Gustafson's law is better, as purpose of parallelization is to be able to solve larger and larger problems as number of processors is increased. Moreover, Gustafson's law predicts that efficiency becomes a constant as number of processors is increased, unlike Amdahl's law which predicts that efficiency keeps decreasing and approaches zero as number of processors is increased. It is observed from the simulations using PMNIM, for fixed problem size per processor, that the scheme in fact follows Gustafson's law as efficiency reaches (almost) a constant as number of processors is increased. The scalability and efficiency are observed to be quite good for the developed computer code.

However, it is noted from the simulations carried out with the variable problem size per processor, Amdahl's law is not strictly followed. The efficiency first increases with the increasing number of processors, achieves a maximum, and then starts decreasing again. This phenomenon, known as superlinear speedup, is seen only if computational load per processor is relatively large. Superlinear speedups have been observed in the past and are associated with memory effects (Gustafson 1990, Helmbold 1989, Parkinson 1986). This behavior is somewhat pronounced in the PMNIM because of large number of variables stored per grid point in this method.

In this dissertation, developed PMNIM is used for simulation of ABC flows. Recently, these flows are studied as the prototype for the development of turbulence (Podvigina 1994, Ashwin 2003). One interesting phenomenon associated with the ABC flows is the heteroclinic cycles. In the present work 1:1:1 ABC flow is studied for the wavenumbers $k = 1$ and $k = 2$. The numerical approximation for the forcing term in the governing equations of the ABC flows, for the two cases, is developed and implemented in order to carry out these simulations.

The results obtained for $k = 1$ are compared with the results available in the literature obtained with the Fourier pseudo spectral method (Podvigina 1994). The grid sizes and time steps used are similar to that used in the cited work. The flow loses its symmetry at higher Reynolds number, as a second set of steady state solutions is obtained. Further increase in the Reynolds number leads to heteroclinic cycles in the flow. The flow becomes fully chaotic at a Reynolds number close to 25. These results match very well with the results given by Podvigina (1994), especially at the higher Reynolds numbers. Some small discrepancies are seen in the results at the lower Reynolds number. However, these discrepancies in the results can be attributed to the fact that the perturbations present in the two approaches are different.

One can expect that for wave number $k = 2$, due to smaller scales present in the flow, the instability and chaos are likely to appear at a Reynolds number lower than that for $k = 1$. This is indeed the case as a second steady state is obtained around a Reynolds number of 4. Unlike the case of $k = 1$, this steady state also has wavenumbers which are smaller than enforced wavenumber ($k = 2$). The abovementioned steady state becomes unstable as Reynolds number is increased and heteroclinic cycles appear in the flow.

Further increase in the Reynolds number results in the heteroclinic cycles which consist of periodic solutions. Moreover, these heteroclinic cycles are separated by intervals of the non-periodic steady state solutions. One interesting phenomenon observed in this flow is the overlapping of different type of heteroclinic cycles, resulting in a very complex evolution of the flow with time. The flow becomes fully chaotic around a Reynolds number of 14. The flow at this and higher Reynolds numbers consist of the heteroclinic cycles of the chaotic invariant sets.

6.2 Future Work

One obvious limitation of the current implementation of the MNIM for turbulent flows is the lack of good solvers for the set of algebraic equations. For the two dimensional convection diffusion equation, NIM formulation has been modified by Toreja (2002) to incorporate *PETSc* (Portable, Extensible Toolkit for Scientific Computation-developed at Argonee National Laboratory) solvers and preconditioner routines. However, there is need for better solvers for non-linear algebraic equations which result when the nodal integral approach is applied to solve the incompressible Navier-Stokes equation. The Guass-Seidel (or SOR) solver with Picard linearization, used in the present work, has obvious limitations for non-linear set of algebraic equations. This is especially true for DNS of turbulent flows because of high degree of non-linearity.

It is interesting to note that unlike applied mathematics community, CFD practitioners have largely preferred Picard linearization with splitting of the equations and operators. However, the applied mathematicians generally use Newton iterations

(Knoll 2003). The advantage of the Newton method is that it has faster nonlinear convergence and is also free from operator splitting error. Among the CFD researchers, Vanka (1983) and MacArthur (1989) were two of the early proponents of the Newton method. The incompressible Navier-Stokes equation has been also solved using Jacobian Free Newton Kyrlov (JFNK) method (e.g. Brown 1990, Johnson 1997). In the future, the JFNK solver can be incorporated in the MNIM computer code. The advantages of JFNK are discussed by Knoll (2003).

The parallelization strategy also needs to be improved especially to include irregular geometries. However, if JFNK type of scheme is used as the solver of non-linear algebraic equations, one essentially needs a parallel JFNK based solver for the set of non-linear algebraic equations. Since, parallel solver for the system of algebraic equations can be developed without the use of domain decomposition paradigm; it is independent of the geometry of the computational domain. In the context of the finite volume based methods, such parallel JFNK solvers have been recently developed for incompressible flows (Gropp 2001). Although, these solvers have been developed for the non-linear algebraic equations resulting from the finite volume based approach, it is not unreasonable to assume that they can be extended to solve such equations arising from the MNIM.

Another possible extension to the existing MNIM is to include capabilities for the simulation of relatively complex physical phenomena. For example, the numerical scheme can be modified to solve magnetohydrodynamic (MHD) equations. The MHD formulation for systems with small magnetic induction (e.g. liquid metals) essentially consists of a Poisson equation for electric potential coupled with the appropriately

modified Navier-Stokes equation. The existing scheme for the pressure (Poisson) equation can be relatively easily modified to develop a numerical scheme for the abovementioned Poisson equation. The coupling of the numerical scheme for the electric potential equation with the existing numerical schemes for the Poisson equation for pressure and the momentum equations (with appropriately modified forcing term) will lead to efficient MNIM for the liquid metal MHD. In contrast, the MNIM for the MHD with significant magnetic induction will be more challenging. Similarly, MNIM can be extended to model multiphysics phenomena such as coupled neutronics-thermal hydraulics for reactor safety analyses.

Recently, there has been significant development in the extension of the 2D NIM for the geometries with curved boundaries, both for the fluid flow (Toreja 2003a) as well as neutron diffusion (Gu 2005) by coupling NIM to other numerical schemes (e.g. finite element method, finite analytic method). Such extension in the 3D will be quite useful. Another approach that can be used to deal with the curved boundaries is the cut cell approach which has been developed in the context of the finite volume methods (e.g. Berger 1989). In the same context, another approach which has been used is to transform irregular quadrilateral nodes to rectangular (or square) nodes through algebraic transformations. The algebraic transformations have been used in the past both for the finite element method (e.g. Barlow 1989) and finite volume method (e.g. Darr 1990). Recently there have been some developments in these transformations for the NIMs for the convection-diffusion equations (Nezami 2007). In the future, these methods can be extended for the fluid flow equations. The extension of 2D MNIM for k-ε modeling to 3D, though not trivial, is relatively straightforward.

References

[Abe 1994] K. Abe, T. Kondoh, Y. Nagano, "A new turbulence model for predicting fluid flow and heat transfer in separating and reattachingflows-l. Flow field calculations", *Int. J. Heat Mass Transfer*, **37(1)**, 139-151, 1994.

[Amdahl 1967] G. M. Amdahl, "Validity of the single-processor approach to achieving large scale computing capabilities", *AFIPS Conference Proceedings*, **30**, 483-485, 1967.

[Arnold 1965] V. I. Arnold, "Sur la topologie des écoulements stationnaires des fluides parfaits", *C. R, Acad. Sci. Paris*, **261**, 17-20, 1965.

[Ashwin 2003] P. Ashwin, O. Podvigina, "Hopf bifurcation with cubic symmetry and instability of ABC flow", *Proc. R. Soc. London*, **459,** 1801–1827, 2003.

[Azmy1983] Y. Y. Azmy, J. J. Dorning, "A nodal integral approach to the numerical solution of partial Differerential Equations", *Advances in Reactor Computations*, **II**, 893–909, 1983.

[Azmy 1985] Y. Y. Azmy, *Nodal method for problems in fluid mechanics and neutron transport*, Ph.D. thesis, University of Illinois, 1985.

[Baldwin 1978] B. S. Baldwin, H. Lomax, "Thin layer approximation and algebraic model for separated turbulent flows", *AIAA Paper 78-257*, 1978.

[Baldwin 1990] B. S. Baldwin, T. J. Barth, "A one-equation turbulence transport model for high Reynolds number wall-bounded flows", *NASA TM 102847*, 1990.

[Barlow 1989] Barlow, J. "More on optimal stress points-reduced integration, element distortion and error estimation", *Int. J. Numer. Meth. Engng*, **28**, 1487-1504, 1989.

[Basu 1994] A. J. Basu, "A parallel algorithm for spectral solution of the three-dimensional Navier–Stokes equations", *Parallel Comput.*, **20**, 1191–1204, 1994.

[Berger 1989] M. Berger, R. LeVeque, "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries", *AIAA Paper 89-1930*, 1989.

[Bouffanais 2007] R. Bouffanais, M. O. Deville, E. Leriche, "Large-eddy simulation of the flow in a lid-driven cubical cavity", *Phys. Fluids*, **19**, 055108, 2007.

[Brown 1990] P.N. Brown, Y. Saad, "Hybrid Krylov methods for nonlinear systems of equations". *SIAM J. Sci. Stat. Comput.*, **11**, 450–481, 1990.

[Cambon 1997] C. Cambon , N. N. Mansour, F. S. Godeferd, "Energy transfer in rotating turbulence",. *J. Fluid Mech.* **337**, 303-322, 1997.

[Chen 1992] S. Chen, X. Shan, "High-resolution turbulent simulations using the connection machine-2", *Comput. Phys.*, **6**, 643–646, 1992.

[Chien 1982] K. Y. Chien, "Predictions of channel and boundary-layer flows with a low Reynolds number turbulence model", *AIAA Journal*, **20(1)**, 33-38, 1982.

[Childress 1970] S. Childress, "New solutions of the kinematic dynamo problem", *J. Math. Phys.* **11**,3063–3967, 1970.

[Darr 1991] J. H. Darr, S. P. Vanka, "Separated flow in a driven trapezoidal cavity", *Phys. Fluids A*, **3(3)**, 385-392, 1991.

[del Alamo 2003] J. C. del Alamo, J. Jimenez, "Spectra of the very large anisotropic scales in turbulent channels", *Phys. Fluids*, **15(6)**, L41-L44, 2003.

[Dombre 1986] T. Dombre, U. Frisch, J. M. Greene, M. Hénon, A. Mehr and A. Soward, "Chaotic streamlines in the ABC flows", *J. Fluid Mech.* **167**, 353–391, 1986.

[Durbin 2001] P. A. Durbin, B. A. P. Reif, *Statistical Theory and Modeling for Turbulent Flows*, John Wiley & Sons Ltd., New York, 2001.

[Elnawawy 1990] O. A. Elnawawy, A. J. Valocchi, A. M. Ougouag, "The cell analytical–numerical method for solution of the advection–dispersion Equation: Two-dimensional Problems", *Water. Resour. Res.*, **26(11)**, 2705–2716, 1990.

[Eswaran 1988] V. Eswaran, S. B. Pope, "An examination of forcing in direct numerical simulations of turbulence", *Comput. Fluids*, **16**, 257-278, 1988.

[Esser 1993] P. D. Esser, R. J. Witt, "An upwind nodal integral method for incompressible fluid flow", *Nucl. Sci. Engrg.*, **114**, 20-35, 1993.

[Field 1996] M. J. Field, *Lectures on Bifurcations, Dynamics and Symmetry (Pitman Research Notes in Mathematics 356)*, Addison-Wesley Longman Ltd., Harlow UK, 1996.

[Fox 1972] D. G. Fox, D. K. Lilly, "Numerical simulation of turbulent flows", *Rev.Geophys.Space Phys*, **10,** 51-71, 1972.

[Galloway 1984] D. J. Galloway, U. Frisch, "A numerical investigation of magnetic field generation in a flow with chaotic streamlines," *Geophys. Astrophys.Fluid Dynamics*, **29**, 13-18, 1984.

[Garg 1997] R. P. Garg, J. H. Ferziger, S. G. Monismith, "Hybrid spectral finite

difference simulations of stratified turbulent flows on distributed memory architectures", *Int J. Numer. Meth. Fluids*, **24**, 1129–1158, 1997.

[Gilbert 1992] A. D. Gilbert, "Magnetic field evolution in steady chaotic flows", *Phil. Trans. Roy. Soc. London*, **339**, 627-656, 1992.

[Gropp 1999a] W. Gropp, A. Skjellum, E. Lusk, *Using MPI: Portable Parallel Programming with the Message-Passing Interface*, MIT Press, Cambridge, MA, 1999.

[Gropp 1999b] W. Gropp, A. Skjellum, R. Thakur, *Using MPI-2: Advanced Features of the Message-Passing Interface*, MIT Press, Cambridge, MA, 1999.

[Gropp 2001] W. D. Gropp, D. K. Kaushik, D. E. Keyes, B. F. Smith, "High performance parallel implicit CFD", *Parallel Comput.*, **27**, 337-362, 2001.

[Gu 2005] Y. Gu, Rizwan-uddin, "A hybrid method for multi-group neutron diffusion equations in arbitrary geometry", *Int. Topical Meeting on Mathematics and Computations, Supercomputing, Reactor Physics and Nuclear and Biological Applications*, on CD ROM, 2005.

[Gustafson1988] J. L. Gustafson, "Reevaluating Amdahl's law", *CACM* **31(5)**, 532-533, 1988.

[Gustafson1990] J. L. Gustafson, "Fixed time, tiered memory, and superlinear speedup", *Proceedings of the Fifth Conference on Distributed Memory Computing (DMCC5)*, **II,** 1255-1260, 1990.

[Harlow 1967] F. H. Harlow, P. I. Nakayama, "Turbulence transport equations", *Phys. Fluids*, **10**, 2323-2332, 1967.

[Helmbold 1989] D. P. Helmbold, C. E. McDowell, "Modeling Speedup(n) greater than n", *Proceedings of International Conference on Parallel Processing*. **III**, 219-295, IEEE Computer Society Press, 1989.

[Hennart 1986] J. P. Hennart, "A general family of nodal schemes", *SIAM J. Sci. Stat. Comp.*, **7(1)**, 264–287, 1986.

[Hénon 1966] M. Hénon, "Sur la topologie des lignes de courant dans un cas particulier", *Comptes Rendus Acad. Sci. Paris* **262**, 312–314, 1966.

[Horak 1985] W. C. Horak, J. J. Dorning, "A nodal coarse-mesh method for the efficient numerical solution of laminar flow problems", *J Comp Physics*, **59**, 405-440, 1985.

[Hoyas 2006] S. Hoyas, J. Jimenez, "Scaling of velocity fluctuations in turbulent channels up to $Re_\tau = 2000$", *Phys. of Fluids*, **18**, 011702, 2006.

[Huang 1995] P. G. Huang, G. N. Coleman, P. Bradshaw, "Compressible turbulent channel flow: DNS results and modelling", *J. Fluid Mech.* **305**, 185-218, 1995.

[Jackson 1991] E. Jackson, Z. S. She, S. A. Orszag, "A case study in parallel computing: I. homogeneous turbulence on a hypercube", *J. Sci. Comput.*, **6**, 27–45, 1991.

[Jiménez 1991] J. Jiménez, P. Moin, "The minimal flow unit in near-wall turbulence", *J. Fluid Mech.* **225**, 221-240, 1991.

[Johan 1992] Z. Johan, T. J. R. Hughes, "A data parallel finite element method for computational fluid dynamics on the connection machine system", *Comput. Meth. Appl. Mech. Eng.*, **99**, 113–134, 1992.

[Johnson 1997] R.W. Johnson, P.R. McHugh, D.A. Knoll, "High-order scheme implementation using Newton–Krylov solution methods", *Numer. Heat Trans., Part B*, **31**, 295-312, 1997

[Jones 1972] W P Jones, B E Launder, "The prediction of laminarization with a two-equation model of turbulence", *Int. J. Heat and Mass Transfer,* **15**, 301-314, 1972.

[Kim 1987] J. Kim, P. Moin, R. D. Moser, "Turbulence statistics in fully developed channel flow at low Reynolds number", *J. Fluid Mech.*, **177**, 133-166, 1987.

[Kleiser 1991] L. Kleiser, T. A. Zang, "Numerical simulation of transition in wall bounded shear flows", *Annu. Rev. Fluid Mech.*, **23**, 495–537, 1991.

[Knoll 2004] D. A. Knoll, D. E. Keyes, "Jacobian-free Newton–Krylov methods: a survey of approaches and applications", *J. Comp. Phys.*, **193(2)**, 357-397, 2004.

[Krupa 1995] M. Krupa, I. Melbourne, "Asymptotic stability of heteroclinic cycles in systems with symmetry", *Ergod. Th. & Dynam. Sys.*, **15**, 121-147, 1995.

[Krupa 1997] M. Krupa, "Robust heteroclinic cycles", *J. Nonlin. Sci.*, **7(2)**, 129-176, 1997.

[Launder 1975] B. E. Launder, G. J. Reece, W. Rodi, "Progress in the development of a Reynolds-stress turbulent closure", *Journal of Fluid Mechanics*, **68(3)**, 537-566, 1975.

[Lawrence 1986] R. D. Lawrence, "Progress in nodal methods for the solutions of the neutron diffusion and transport equations", *Progress in Nuclear Energy*, **17(3)**, 271-301, 1986.

[Le 1992] H. Le, P.Moin, "Direct numerical simulation of turbulent flow over a backward-facing step", *Rep. TF-58*, Thermosci. Div., Dept. Mech. Eng., Stanford Univ., Calif., 1992.

[Leriche 2000] E. Leriche, S. Gavrilakis, "Direct numerical simulation of the flow in the lid-driven cubical cavity," *Phys. Fluids*, **12**, 1363-1376, 2000.

[Levit 1988] C. Levit, D. Jespersen, "Explicit and implicit solution of the Navier–Stokes equations on a massively parallel computer", *Comput. Struct.*, **30**, 385–393, 1988.

[Liu 2003] C. H. Liu, D. Y. C. Leung, C. M. Woo, "Development of a scalable finite element solution to the Navier-Stokes equations", *Computational Mechanics*", **32(3)**, 185-198, 2003.

[Lloyd 2005] D. J. B. Lloyd, A. R. Champneys, R. E. Wilson, "Robust heteroclinic cycles in the one-dimensional complex Ginzburg–Landau equation", *Physica D*, **204**, 240-268, 2005.

[MacArthur 1989] W. MacArthur, S.V. Patankar, "Robust semidirect finite difference methods for solving the Navier–Stokes and energy equations", *Int. J. Numer. Methods Fluids*, **9**, 325–340, 1989.

[Melbourne 1989] I. Melbourne, "Intermittency as a codimension three phenomenon", *Dyn. Diff. Eqn.*, **1**, 347-367, 1989.

[Mercader 2001] I. Mercader, J. Prat, E. Knobloch, "Robust heteroclinic cycles in two-dimensional Rayleigh-Bénard Convection without Boussinesq symmetry", *International Journal of Bifurcation and Chaos*, **12 (11)**, 2501-2522, 2001.

[Michael 2001] E. P. E. Michael, J. J. Dorning, Rizwan-uddin, "Studies on nodal integral methods for the convection–diffusion heat equation", *Nucl. Sci. Engrg.*, **137**, 380–399, 2001.

[Mittal 1995] S. Mittal, T. E. Tezduyar, "Parallel finite element simulation of 3D incompressible flows: Fluid-structure interactions", *Int. J. Numer. Meth. Fluids*, **21(10)**, 933-953, 1995.

[Moin 1998] P. Moin, K. Mahesh, "Direct numerical simulation: A tool in turbulence research," *Annu. Rev. Fluid Mech.*, **30**, 539-578, 1998.

[Morton 1994] K. W. Morton, D. F. Mayers, *Numerical Solutions of Partial Differential Equations*, Cambridge University Press, 1994.

[Moser 1987] R. D. Moser, P. Moin, "The effects of curvature in call bounded turbulent flows", *J. Fluid Mech.*, **175**, 479-510, 1987.

[Nagano 1990] Y. Nagano, M. Tagawa, "An improved k-epsilon model for boundary layer flows", *Journal of Fluids Engineering*," **112**, 33-39, 1990.

[Naik 1993] H. Naik, V. K. Naik, M. Nicoules, "Parallelization of a class of implicit

finite difference schemes in computational fluid dynamics", *Int. J. High Speed Comp.* **5**, 1-50, 1993.

[Nezami 2007] E. G. Nezami, S. Singh, N. Sobh, Rizwan-uddin, "A formulation for incorporation of irregular-shape elements in nodal integral method", *Submitted to Int J Numer Meth Fluid* , 2007

[Nonino 1988] C. Nonino, S. D. Giudice, "Finite element analysis of turbulent forced convection in lid-driven rectangular cavities", *Int.J. Numer. Meth. Engng*., **25**, 313-329, 1988.

[Park 1994] T. S. Park, H. J. Sung, "A nonllinear low-Reynolds-number model for turbulent separated and reattaching flows: Flow field computations", *Int. J. Heat Mass Transfer*, **38(14)**, 2657-2666, 1994.

[Parkinson 1986] D. Parkinson, "Parallel efficiency can be greater than unity", *Parallel Comput*, **3**, 261-262, 1986.

[Passoni 2001] G. Passoni, P. Cremonesi, G. Alfonsi, "Analysis and implementation of a parallelization strategy on a Navier-Stokes solver for shear flow simulations", *Parallel Comput.*, **27(13)**,1665-1685, 2001.

[Pelz 1991] R. B. Pelz, "The parallel Fourier pseudo-spectral method", *J. Comput. Phys.*, **92**, 296–312, 1991.

[Podvigina 1994] O. Podvigina, A. Pouquet, "On the non-linear stability of the 1:1:1 ABC flow", *Physica D*, **75**, 471–508, 1994.

[Podvigina 1999] O. Podvigina, "Spatially-periodic steady solutions to the three-dimensional Navier–Stokes equation with the ABC-force", *Physica D*, **128**, 250-272, 1999.

[Podvigina 2006] O. Podvigina, P. Ashwin, D. Hawker, "Modelling Instability of ABC Flow using a Mode Interaction between Steady and Hopf Bifurcations with Rotational Symmetries of the Cube", *Physica D*, **215**, 62-79, 2006.

[Prestin 1995] M. Prestin, L. Shtilman, "A parallel Navier–Stokes solver: the Meiko implementation", *J. Supercomput.*, **9**, 347–364, 1995.

[Riley 1986] J. J. Riley, R. W. Metcalfe, and S. A. Orszag, "Direct numerical simulations of chemically reacting mixing layers", *Phys. Fluids*, **29**(2), 406-422, 1986.

[Rodi 1993] W. N. Rodi, N. Mansour, "Low Reynolds number k-epsilon modeling with the aid of direct simulation data", *Journal of Fluid Mechanics*, **250**, 509-529, 1993.

[Rogallo 1981] R.S. Rogallo, "Numerical experiments in homogeneous turbulence",

NASA-TM-81315, 1981.

[Schmid 2001] P. J. Schmid, D. S. Henningson, *Stability and Transition in Shear Flows*, Springer, Cambridge University Press, London, 2001.

[Smith 1967] A. M. O. Smith, T. Cebeci, "Numerical solution of the turbulent boundary layer equations", *Douglas aircraft division report DAC 33735*, 1967.

[Spalart 1992] P. R. Spalart, and S. R. Allmaras, "A one-equation turbulence model for aerodynamic flows", *AIAA Paper 92-0439*, 1992

[Suzuki 1992] Y. Suzuki, N. Kasagi, "Evaluation of hot wire measurements in wall shear turbulence using a direct numerical simulation database", *Exp. Therm. Fluid Sci.*, **5**, 69-77, 1992.

[Tannehill 1997] J. C. Tannehill, D. A. Anderson, R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer* (2^{nd} Ed.), Taylor and Francis, 1997.

[Toreja 2002] A. J. Toreja, *A nodal approach to arbitrary geometries, and adaptive mesh refinement for the nodal method*, Ph.D. thesis, University of Illinois, 2002.

[Toreja 2003a] A. J. Toreja, Rizwan-uddin, "A hybrid nodal method for time-dependent incompressible flow in two-dimensional arbitrary geometries", *Nuclear Mathematical and Computational Sciences: A Century in Review, A Century Anew*, on CD ROM, 2003.

[Toreja 2003b] A. J. Toreja, Rizwan-uddin, "Hybrid numerical methods for convection–diffusion problems in arbitrary geometries", *Computers & Fluids*, **32(6)**, 835-872, 2003.

[Vanka 1983] S.P. Vanka, G.K. Leaf, "Fully coupled solution of pressure-linked fluid flow equations", *Technical Report ANL-83-73*, Argonne National Laboratory, 1983

[Wakashima 2003] S. Wakashima, T. S. Saitoh, "Benchmark solutions for natural convection in a cubic cavity using the high order time-space method", *Int.. J.. Heat Mass Transfer*, **47**, 853-864, 2003.

[Wang 2003] F. Wang, Rizwan-uddin, "A modified nodal scheme for the time-dependent, incompressible Navier-Stokes equations", *J. Comp. Physics.*, **187**,168-196, 2003.

[Wang 2005] F. Wang, Rizwan-uddin, "Modified nodal integral method for the three-dimensional time-dependent incompressible Navier-Stokes equations", *Nucl. Sci. Engrg.*, **149**, 107-114, 2005.

[Wasfy 1998] T. Wasfy, A. C. West, V. Modi, "Parallel finite element computation of unsteady incompressible flows", *Int. J. Numer. Meth. Fluids*, **26**, 17–37, 1998.

[Wescott 2001] B. Wescott, Rizwan-uddin, "An efficient formulation of the modified nodal integral method and application to the two dimensional Burgers equation", *Nucl. Sci. Engrg.*, **139**, 293–305, 2001.

[Wilcox 1988] D. C. Wilcox, "Re-assessment of the scale-determining equation for advanced turbulence models", *AIAA Journal*, 26, 1414-1421, 1988.

[Wilson 1988] G. L. Wilson, R. A. Rydin, Y. Y. Azmy, "Time-dependent nodal integral method for the investigation of bifurcation and nonlinear phenomena in fluid flow and natural convection", *Nucl. Sci. Engrg.*, **100**, 414–425, 1988.

[Wu 1997] J. Wu, H. Ritzdorf, K. Oosterlee , B. Steckel, A. Schüller, "Adaptive parallel multigrid solution of 2D incompressible Navier–Stokes equations", *Int. J. Numer. Meth. Fluids*, **24**, 875–892, 1997.

[Yakhot 1992] V. Yakhot, S. A. Orszag, S. Thangam, T. B. Gatski, C. G. Speziale, "Development of turbulence models for shear flows by a double expansion technique", *Physics of Fluids A*, **4(7)**, 1510-1520, 1992.

Author's Biography

Suneet Singh was born in Kanpur, India on September 12, 1976. He graduated from Maulana Azad College of Technology (Regional Engineering College), Bhopal, India in 1998 with Bachelor of Engineering in Electrical Engineering. He completed his Master of Technology in Nuclear Engineering and Technology from Indian Institute of Technology, Kanpur, India. He joined the University of Illinois at Urbana-Champaign to pursue Doctor of Philosophy in Nuclear Engineering in 2001. He completed his Ph.D. in 2007.