NON–INVERTED SKEW UPWIND SCHEME FOR NUMERICAL HEAT TRANSFER AND FLUID FLOW SIMULATIONS

By

Emmanuel Olakunle Busayo OGEDENGBE

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES OF
THE UNIVERSITY OF MANITOBA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF MECHANICAL AND MANUFACTURING ENGINEERING
UNIVERSITY OF MANITOBA
WINNIPEG, MANITOBA
AUGUST 2006

UNIVERSITY OF MANITOBA
DEPARTMENT OF
MECHANICAL AND MANUFACTURING ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "**Non–Inverted Skew Upwind Scheme for Numerical Heat Transfer and Fluid Flow Simulations**" by **Emmanuel Olakunle Busayo OGEDENGBE** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dated: August 2006

External Examiner: _____
Manohar Kulkarni, Ph.D, PE

Research Supervisor: _____
Greg F. Naterer, Ph.D, P.Eng

Examining Committee: _____
Arvind H. Shah, Ph.D, P.Eng

_____
Robert W. Derksen, Ph.D, P.Eng

UNIVERSITY OF MANITOBA

Date: **August 2006**

Author: **Emmanuel Olakunle Busayo OGEDENGBE**

Title: **Non–Inverted Skew Upwind Scheme for Numerical Heat Transfer and Fluid Flow Simulations**

Department: **Mechanical and Manufacturing Engineering**

Degree: **Ph.D.** Convocation: **October** Year: **2006**

To the Glory of God.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis studies advection modeling for heat transfer and fluid flow problems using a new Non–Inverted Skew Upwind Scheme (called NISUS). Variants of the new scheme are formulated and developed with 8-noded hexahedral elements using the Finite Element Method (FEM) and rectangular elements based on a Finite Volume Method (FVM). A new method of mass weighting to predict convective fluxes of each scalar from the nodal point values is developed. Due to an explicit representation in terms of nodal variables, local inversion of the upwind coefficient matrix is not needed. Also, this thesis evaluates two variants of the new scheme (i.e., 3-node / 3-point and 4-node / 8-point formulations) within a 3–D FEM and a third variant within a 2–D FVM. The 3–D FEM variants are applied to a variety of test problems involving the transport of a scalar variable, while the 2–D FVM variant is applied to fluid flow problems including natural convection in an enclosure and micro–channel flow simulations. The promising performance of NISUS, as compared with exact and previous solutions, is demonstrated both in terms of accuracy and stability. Furthermore, a new data storage format called Compressed Banded Data (CBD) is developed for sparse banded matrices generated by the control volume finite element method (CVFEM). The platform of the new CBD structure permits dynamic switching between various solvers, without any procedural change in the implementation of existing simulation software. The performance of different Krylov techniques with an ILU(0) preconditioner is observed and compared in three test problems with a direct solver.

Acknowledgements

I would like to thank my supervisor, Dr. G.F. Naterer, for his motivation, suggestions and support during this research. His encouragement and interest provided the motivating energy towards the completion of the work. I am also thankful to Dr. A. Shah and Dr. R.W. Derksen of the University of Manitoba for their contributions within the advisory committee. I appreciate the support of Dr. M.A. Rosen, Dean of the Faculty of Engineering and Applied Science, for the privilege of conducting research at the University of Ontario Institute of Technology (UOIT), Oshawa.

This research is supported through the following sources of funding: NSERC research grant to Dr. G.F. Naterer; University of Manitoba Graduate Fellowship; Edward R. Toporeck Graduate Fellowship in Engineering; and financial support through teaching/research assistantships from the Department of Mechanical and Manufacturing Engineering, University of Manitoba and a research assistantship from UOIT. Also, I am privileged to work in the company of colleagues with great ideas and valuable intellectual capabilities at the University of Manitoba. These include Stephen Glockner, Marko Milanez, 'Sola Adeyinka, Xili Duan, Dapeng Wang, Samuel Paul, and Xin Wang.

Of course, I am grateful to my wife and daughter, Theresa and Christwill, for their moral support and *love*. I also appreciate the prayers and patience of my mother, Mrs. Bernice Ogedengbe, including all my siblings. Without them, this work would never have come into existence (literally).

Winnipeg, Manitoba E.O.B. Ogedengbe

August 10, 2006

Abbreviations

Bi–CGSTAB	.....Bi–conjugate Gradient Stabilized method
CVFEM	.....Control Volume Finite Element Method
CPU	.....Central Processing Unit
CBD	.....Compressed Banded Data
CDS	.....Central Difference Scheme
CLDA	.....Conservative Low-Dispersion Algorithm
CSR	.....Compressed Sparse Row
CSC	.....Compressed Sparse Column
CFD	.....Computational Fluid Dynamics
EDS	.....Exponential Differencing Scheme
FVM	.....Finite Volume Method
FEM	.....Finite Element Method
GMRES	.....Generalized Minimal Residual method
JDS	.....Jagged Diagonal Storage
MEMS	.....MicroElectroMechanical Systems
NISUS	.....Non-Inverted Skew Upwind Scheme
PWIM	.....Pressure Weighted Interpolation Method
QUICK	.....Quadratic Upwind Interpolation for Convective Kinematics
RAM	.....Random Access Memory
SUCCA	.....Skew Upwind Corner Convection Algorithm
TDMA	.....Tri-Diagonal Matrix Algorithm
UDS	.....Upwind Difference Scheme

Nomenclature

a	half-thickness of the parallel plates (m)
aip, bip, cip, dip	boundary integration points
c_p	specific heat capacity (J/kgK)
C	convective length scale
D	diffusion length scale
Gr	Grashof number
ib, jb	first index along x,y-coordinates
ie, je	last index along x,y-coordinates
k	thermal conductivity (W/mK)
kn	Knudsen number
L	length (m)
$\dot{M}$	total mass-flow rates (kg/s)
$\dot{m}_e$	mass-flow rate through the east face (kg/s)
$\dot{m}_n$	mass-flow rate through the north face (kg/s)
p^*	estimate of pressure (N/m^2)
p'	corrected pressure level (N/m^2)
p	current pressure level (N/m^2)
Pr	Prandtl number
Pe	Peclet number
$\mathbf{q}$	flux (per unit area)
Q	flow rate (e.g J/s)
Ra	Rayleigh number
Re	Reynolds number

s, t, u	local coordinates
$\dot{S}'''$	Source term
$\hat{S}$	volumetric source term
t	time (s)
Δt	time step (s)
u^*	estimate of u-velocity (m/s)
u'	corrected u-velocity field (m/s)
u	corrected u-velocity field (m/s)
u_x	velocity component in the direction of flow (m/s)
u_c	centerline velocity (m/s)
u_o	uniform entrance velocity (m/s)
v^*	estimate of v-velocity (m/s)
v'	correction v-velocity field (m/s)
v	corrected v-velocity field (m/s)
$\mathbf{v}$	fluid velocity vector (m/s)
x, y, z	cartesian coordinates (m)

Greek

α	convective coefficient
β	diffusion coefficient
δ	velocity boundary layer thickness (m)
δx	streamwise grid spacing (m)
λ	mean free path (m)
Γ	coefficient of diffusion
μ	viscosity of the fluid (Ns/m^2)
ϕ	scalar field variable
θ	angle of cavity tilt
ρ	density of the fluid (kg/m^3)
σ	accommodation coefficient
ξ_1, ξ_2	slip-coefficients

Subscripts

e, w, n, s	east, west, north, south faces
E, W, N, S	east, west, north, south points
P	control volume point
ip	integration point
up	upstream point

Chapter 1

Introduction

1.1 Background

Numerical modeling of heat transfer and fluid flow, particularly in regards to accuracy and stability, has often been subjectively based on compromise [1, 2, 3, 4, 5]. The skew-upwind scheme for convection modeling has been widely used in numerical techniques for thermal and fluids engineering problems [1, 4, 6, 7]. It has importance in many practical applications. For example, applications appear in microfluidics design optimization [8] and pollution problems [9], among others. Also, its importance appears in various physical phenomena, ranging from single phase to multiphase flows with droplets [10, 11, 12].

Reducing simulation time and numerical error represents an ongoing challenge in the development of next-generation simulation softwares. In some cases, time–consuming features of an algorithm, such as a staggered grid with segregated equations [13], can be re-formulated, i.e., collocated grid with simultaneous solution of non-linear equations [4]. Naterer and Camberos [12] reported that entropy and the

Second Law can provide a systematic way of reducing numerical error, while eliminating instabilities such as numerical overshoots. For example, entropy based diffusion terms have been used to identify the required amount of numerical dissipation in compressible flow analysis [14]. Such entropy based methods can be applied to the overall formulation, or to selected components of the algorithm such as convective upwinding [10], in order to more clearly identify the detailed source of numerical error.

Simulation time and accuracy are major factors characterizing the performance of predictive tools for numerical heat transfer. If the CPU run–time can be appreciably reduced without sacrificing accuracy, then a worthwhile contribution can be realized. Faster thermal simulations can give companies a competitive advantage in various ways. More cases of added complexity could be studied and optimized, thereby allowing the simulation tool to be more closely integrated into the design cycle. If a numerical simulation is too time-consuming in an industrial setting, then it may not be fully accommodated within a tight product development schedule. Also, the time taken to bring the product to market may be lengthened, thereby adversely affecting the company's profit margins.

Numerical simulations of three–dimensional convective heat transfer problems frequently have large computer storage and CPU run–time requirements. In many instances, thermal designers wish to perform heat transfer simulations involving as many grid points as possible with their computer. A majority of computer storage is generally taken with coefficient matrices involving nodal variables of temperature, pressure and velocity. If these matrices exceed the cache memory or RAM limitations of a computer, the CPU run-time can be significantly increased. This thesis attempts to develop a new strategy for reducing storage requirements of 3–D heat transfer

simulations, based on a new compressed banded data (CBD) format. In particular, storage requirements of banded matrices generated from a CVFEM (Control Volume Based Finite Element Method; [10, 15, 16]) are investigated with CBD.

1.2 Convection Modeling

Convection modeling is a time–consuming portion of flow simulations. In finite volume schemes, predictions are needed for the upwind value crossing the edge of a control volume. UDS (Upwind Differencing Scheme) uses the nodal value on the upstream side of the control volume, while CDS (Central Differencing Scheme) takes an average value between upstream and downstream nodes (Patankar, [13]). Since the weighting factor depends on the relative magnitudes of advection and diffusion, hybrid schemes such as EDS (Exponential Differencing Scheme) use the local Peclet number in this weighting. It is well known that UDS leads to false diffusion errors when the flow direction is not closely aligned with the grid lines. Skew upstream differencing (Raithby, [7]) and CLDA (Conservative Low-Dispersion Algorithm; PHOENICS, [17]) have been shown to reduce such false diffusion errors by addressing the flow directionality. Carey et al. [18] have reduced false diffusion errors by interpolating values from upwind corner cells in SUCCA (Skew Upwind Corner Convection Algorithm). However, higher accuracy came at the expense of about 20 per cent additional CPU time during their simulations. This thesis attempts to include corner nodes in a 4-node / 8-point scheme, while simultaneously reducing CPU run-time requirements.

Mass weighted differencing can be effectively applied to convective skew upwinding (Hassan et al., [6]). Muir and Baliga [2] show that flow–oriented interpolation with four-noded tetrahedral elements becomes sensitive to the local Peclet number

and flow directionality. Despite its benefits, directional upwind differencing may lead to other instabilities involving numerical dispersion (Sharif, Busnaina, [19]). Numerical dispersion can be reduced in convection modelling, but often at the expense of upstream values exceeding their physical bounds (Fromm, [20]), thereby producing non–physical results (PHOENICS, [17]). Zhu and Rodi [21] developed a bounded upwind formulation, while exhibiting low dispersion. It is essential that certain upwind variables, such as turbulence kinetic energy or species concentration, remain bounded within physical limiting values. Darwish and Moukalled [22] address modeling of skew upwind schemes to ensure boundedness. Gaskell and Lau [23] develop curvature based corrections for convective upwinding, so that boundedness is preserved. Smith, Hutton [5] and Albada et al. [24] compare various computational methods, while reporting that non-linear convective upwinding may produce non-physical results when boundedness is not preserved (PHOENICS, [17]). Conventional upwind schemes, such as UDS, exhibit first-order accuracy and they may violate local entropy-stability criteria (Naterer, [25]).

Second–order methods such as QUICK (Leonard, [26]) use quadratic interpolation, involving values at three adjacent nodal points, to reduce errors related to numerical diffusion. Agrawal [27] developed a third-order upwind scheme for viscous flows. But a compromise is needed between higher order accuracy and the resulting CPU run-time. For example, a lower order scheme with more elements can often out–perform a higher order scheme for a given CPU run–time. Also, third-order accuracy may sacrifice boundedness in the convective upwinding (Leonard et al., [28]; PHOENICS, [17]). Second-order accuracy with positive coefficients for numerical stability has been reported by Schneider and Raw [4]. But if both nodal and integration point variables are used implicitly for convective upwinding, costly sub–element inversion

of the upwind coefficient matrices is needed. Convergence acceleration represents another significant portion of simulation time.

Hutchinson and Raithby [29] utilize a multigrid method, whereby coarse and fine grids are used together to accelerate solution convergence and reduce solution errors. Alternatively, entropy and the Second Law offer useful ways of establishing error indicators, either by an apparent entropy production difference (Adeyinka, Naterer, [30]) or a weighted entropy residual [14]. Reduced error often leads to improved solution convergence and less simulation time. Such time and accuracy goals become particularly significant when predicting 3-D flows, such as turbomachinery flows (Kictley, [31]) or multi-phase flows [10].

1.3 Past Skew-Upwind Schemes

The skew-upwind method is a widely used technique of convection modeling. Raithby [7] proposed the skew upstream scheme as an alternative to the conventional upstream difference scheme, in order to reduce false diffusion errors in the region of flow where the computational gridline and flow streamlines are not closely aligned. The scheme was formulated with a finite differencing technique and it formed the basis for subsequent developments. Also within the finite volume context, Hassan et al. [6] proposed a mass-flow-weighted skew upwind scheme as an improvement over the conventional skew–upwind scheme. The scheme ensures a reduction of numerical instability and numerical diffusion errors.

Instability problems, leading to numerical oscillations of overshoots and undershoots, result from negative coefficients in the algebraic matrix [4]. Muir and Baliga

[2] presented a 3–D formulation based on tetrahedral elements with flow-oriented interpolation functions, which are sensitive to the local Peclet number and the direction of the local–average velocity vector. Their 3-D work is an extension of the 2–D formulation of Baliga and Patankar [3]. The assumption of an average velocity in the direction of the flow over the element simplifies the implementation of the scheme. However, this raises uncertainty about continuity of the scalar across the bounding surfaces of the tetrahedral elements (especially at high Peclet numbers).

In the finite element context, Schneider and Raw [4] proposed an upwind procedure that accounts for the directionality of the flow field through a skewed approach, while simultaneously precluding the possibility of negative coefficients. They recommended both nodal and integration point values in the approximation of the convected value at the integration point, in order to avoid negative coefficients, especially in a highly non-uniform flow field. The 2–D work was formulated with linear, quadrilateral elements. Although its 3–D extension appears straightforward, the CPU time requirement to perform inversion of 12×12 local elemental matrices may become uneconomical, especially if a more cost-effective procedure could offer the same or comparable accuracy and stability. Local inversion is required when convection upwind variables include both integration point and nodal variables, since each integration point variable must be written explicitly in terms of nodal variables alone.

In this thesis, a new convective upwind scheme called NISUS (Non–Inverted Skew Upwind Scheme) is developed. The main benefit of the proposed schemes lies in avoiding the costly inversion of the upwind coefficient matrix, without any significant loss of accuracy. A detailed description of the overall formulation, including advection, diffusion, transient and source terms, is outlined in sections 3.2 and 3.3. Also, the

implementation of the boundary conditions, which complete the control volume formulation, is described for the 3-D formulation. The description of the 2-D staggered grid finite volume formulation based on NISUS is presented in chapter 4.

1.4 Selecting Direct or Indirect Solvers

For many field problems involving heat transfer and fluid flow, direct solvers offer good results in terms of speed and accuracy. Based on the Gaussian elimination procedure, the direct solution of the set of linear equations can be obtained. Past applications of direct solvers have been studied in several heat transfer applications, such as control volume finite element simulations [15, 32, 33].

Three–dimensional simulations typically involve a large number of algebraic equations in practical industrial applications, often up to tens or hundreds of thousands of equations. In these cases, direct solvers become impractical for large-scale simulations, while other alternatives such as iterative methods based on Krylov subspace techniques, become more cost-effective [34, 35]. Applications of these iterative methods to large-scale problems require effective preconditioning of the coefficient matrices. Furthermore, the performance of the iterative solver depends on the type of preconditioning and the method of data storage.

In convection problems, numerical methods often utilize segregated algorithms, such as SIMPLE [2, 13] or splitting (projection) methods [36], together with iterative solvers such as Krylov subspace methods [37, 38]. Finite volume discretization of the heat transfer and fluid flow equations, based on a segregated algorithm, generally yields a diagonally dominated matrix system, specifically a tri–diagonal matrix system. An example of this type of solver, known as the Tri-Diagonal Matrix Algorithm

(TDMA), uses line Gauss-Seidel relaxation. This type of matrix structure is readily handled by conventional iterative solvers [8, 39]. However, a simultaneous solution of the momentum, mass and energy equations yields a coupled multi-variable system of equations. Pressure and velocity fields appear simultaneously in the momentum equations. Handy et al. [40] have presented a comparison between coupled and segregated solvers for fluid flow problems. In addition to differences involving coupled and segregated methods, other notable differences arise between matrix structures in various schemes. In this thesis, the performance of preconditioned solvers and data compression will be studied, particularly in regards to banded matrices generated by a CVFEM (Control Volume Finite Element Method).

1.5 Data Storage Formats

Computational effort in numerical simulations is largely dependent on the data storage procedure. Effective manipulation of non-zero entries within a sparse matrix has fundamental significance in the design of iterative solvers. Many past storage formats have been developed for sparse matrix systems, including the Compressed Sparse Row (CSR) format, Jagged Diagonal Storage (JDS) format and the Compressed Sparse Column (CSC) format [41, 42, 43]. These methods attempt to eliminate unnecessary storage of zero entries. Each compression format is closely linked with a particular matrix structure and numerical algorithm. As a result, past compression algorithms cannot be directly applied to hybrid methods such as a CVFEM, due to the unique banded sparse matrices generated by such methods. Examples of unique features in a CVFEM include a certain type of diagonal dominance and problem-dependent coefficient structure. In view of shortcomings of existing data compression algorithms,

a new compressed banded data (CBD) format will be developed for a CVFEM in this thesis.

1.6 Application Problems

In this thesis, new methods of numerical modeling will be developed and applied to practical application problems. For example, these problems include free convection in tilted enclosures and microchannel flows. Natural convection in an enclosed vertical rectangular cavity with differentially heated side walls has been the subject of many experimental, analytical and numerical studies. In this thesis, natural convection in tilted enclosures will be studied. It involves varying components of the gravitational field in the vertical direction.

Hart [44] carried out a systematic study of natural convection in a cavity. His study entailed a large aspect ratio cavity. Results showed that the type of fluid circulation was a function of $Ra \sin\theta$ alone for a fixed Prandtl number. Also, the work studied unsteadiness and turbulence structures that characterize high Rayleigh number flow. However, the study did not include heat transfer analysis. In this thesis, different correlations will be proposed to predict the heat transfer across a cavity at various orientations.

Another application problem in this thesis is microchannel flows. Microsystems based on MEMS (i.e., MicroElectroMechanical Systems) are capable of sensing and controlling physical processes with length scales on the order of one micron [45, 46]. Microchannel flows are divided into different flow regimes of CFD (Computational Fluid Dynamics), based on the ratio of the mean free path of the fluid to the characteristic length scale known as the Knudsen number (Kn). At $10^{-3} < Kn < 10^{-1}$, the flow is within the slip flow regime where the continuum-based equations (i.e.,

Navier-Stokes Equations) can be used with a slip boundary condition.

For gas flows, the dynamics associated with MEMS may involve effects of rarefaction and compression. The study of rarefied flow in channels has attracted past interest in numerous studies. Prud'homme et al. [47] developed a 1-D model that uses perturbation predictions of gaseous flow in a microchannel, while neglecting radial pressure gradients. That work was motivated by an attempt to analyze a rigorous method by which a capillary flow viscometer can be calibrated for gaseous viscosity measurements.

Using a 2-D cartesian geometry, Harley et al. [48] investigated gas flow in microchannels at various Knudsen numbers. Their work comprised both experimental and analytical flow predictions with rarefied effects and negligible wall normal pressure gradients. With a first-order correction of the non-zero wall velocity for slightly rarefied flow, their study demonstrated the contribution of non-zero slip on the mass-flow pressure-drop relationship. Similar observations were confirmed by Arkilic et al. [49], with additional data regarding mass flow rates. Using a perturbation expansion of the Navier-Stokes equations, significant contributions of compressibility and rarefied effects were noted. Additional pressure distribution studies were highlighted by Pong et al. [50], although corresponding data on mass-flow rates was not demonstrated.

Apart from these analytic and experimental investigations, numerical studies of microchannel flows have generally used either the continuum [51] or molecular approach [52]. The continuum approach solves transport equations for macroscopic fluid properties (dependent variables) as a function of the spatial coordinates, while the molecular approach considers individual particle dynamics based on a Boltzmann distribution at the temperature of interest. This theses focuses on a continuum

approach, rather than molecular study of gas flow. Also, relevant implications of micro-scale transport near the boundary will be studied.

This chapter has conducted a literature review of past studies related to convection models and iterative solvers. In the next chapter, the numerical formulation of differential equations governing heat transfer and fluid flow problems will be described. Chapter 3 develops new upwind schemes using a finite element method, while chapter 4 develops a variant of the new scheme for the finite volume method. In chapter 5, the iterative solver and data storage methods for the new schemes03 are developed. Numerical results from these schemes are presented in chapter 6. Finally, conclusions and recommendations for future research are presented in chapter 7.

Chapter 2

Numerical Formulation

In this chapter, descriptions of the numerical procedure of formulating both heat transfer and fluid flow problems are presented. This includes discretization of the governing equations, domain and desired elements in the formulation.

2.1 Introduction

Engineering analysis of heat transfer and fluid flow often requires types of design tools. Analytical methods of design usually depend on simplifying assumptions, in order to make the problem tractable. On the other hand, experimental methods can produce more reliable physical data. However, experiments can be prohibitively expensive and time consuming. Since the advent of digital computers, a computational approach of solving complex problems gives an alternative effective tool for product development.

In the numerical formulation of thermofluid problems, governing equations are solved over a solution domain, subject to certain boundary conditions. The formulation involves discretization of terms in the governing equations. In this thesis, these terms will include the transient term, advection term, diffusion term and source term (including viscous diffusion and pressure gradient terms for fluid flow problems).

Conventional methods for such discretization include the finite difference method [53], finite volume method [13] and the finite element method [54].

In this thesis, both the Control Volume Finite Element Method (CVFEM) and the staggered grid Finite Volume Method (FVM) will be studied. The algebraic equations describing the system are obtained through balances of the conserved quantities (e.g. mass, energy, and momentum). Notable advantages of these approaches have been documented previously [55]. In this thesis, a new convection scheme will be developed for the control volume formulation. It will be developed to reduce requirements of CPU run-time [56] and memory [57] for heat transfer and fluid flow simulations.

2.2 Governing Equations

The three–dimensional scalar transport equation for transient advection–diffusion problems is written as

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla \cdot (\rho \mathbf{v}\phi) + \nabla \cdot (\Gamma \nabla \phi) = \hat{S} \tag{2.2.1}$$

In this equation, ϕ is a general scalar variable (such as temperature or concentration of a pollutant in an air stream), ρ is the mass (fluid) density, $\mathbf{v}$ is the fluid velocity vector, Γ is the diffusion coefficient, and $\hat{S}$ is the volumetric generation rate or a source/sink term.

The mass, momentum and energy equations are special cases of Eqn. (2.2.1) and they can be written as

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) = 0 \tag{2.2.2}$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho v u) = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[\mu\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \dot{S}_u''' \tag{2.2.3}$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v v) = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial y}\left[\mu\left(\frac{\partial v}{\partial x} + \frac{\partial v}{\partial y}\right)\right] + \dot{S}_v''' \tag{2.2.4}$$

$$\rho c_p \frac{\partial}{\partial t}(T) + \frac{\partial}{\partial x}(\rho c_p u T) + \frac{\partial}{\partial y}(\rho c_p v T) = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) \tag{2.2.5}$$

Using the Boussinesq approximation for free convection problems,

$$\begin{aligned} \dot{S}_u''' &= \rho_\infty \beta_\infty g_x (T_p^u - T_\infty) \\ \dot{S}_v''' &= \rho_\infty \beta_\infty g_y (T_p^v - T_\infty) \end{aligned} \tag{2.2.6}$$

In these equations, the source terms ($\dot{S}_u'''$ and $\dot{S}_v'''$) incorporate the effects of gravity (g_x and g_y), coefficient of volume expansion (β_∞) and the temperature difference between the center of each control volume (T_p^u or T_p^v) and the reference temperature (T_∞).

2.3 Domain and Control Volume Discretization

A typical three–dimensional problem domain is illustrated in Fig. 2.1. The domain is sub–divided into smaller sub–domains by 8–noded hexahedral elements (Fig. 2.2) where a local coordinate system (s_1, s_2, s_3) is defined. In Fig. 2.2, the first sub–control volume is shown with its three integration points, $ip2$, $ip7$, and $ip9$. For an individual element, the local node numbers range from 1 through 8. The global coordinate (x, y, z) of any point and the dependent scalar, ϕ, within the element are expressed in the following form,

$$\phi = \sum_{i=1}^{8} N_i \phi_i \tag{2.3.1}$$

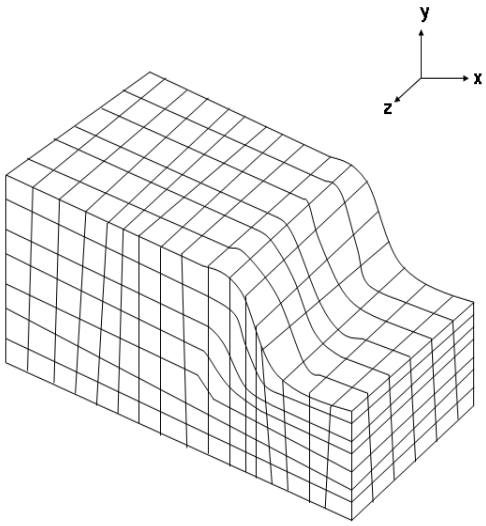


Figure 2.1: Typical three–dimensional domain

$$x = \sum_{i=1}^{8} N_i x_i \tag{2.3.2}$$

$$y = \sum_{i=1}^{8} N_i y_i \tag{2.3.3}$$

$$z = \sum_{i=1}^{8} N_i z_i \tag{2.3.4}$$

where the shape functions are N_i and x_i, y_i, z_i, and ϕ_i are the nodal values of x, y, z, and ϕ at nodes 1 to 8 for i ranging from 1 to 8. The eight–noded hexahedral shape functions are given by [32]

$$N_1 = \frac{1}{8}(1 + s_1)(1 + s_2)(1 + s_3) \tag{2.3.5}$$

$$N_2 = \frac{1}{8}(1 - s_1)(1 + s_2)(1 + s_3) \tag{2.3.6}$$

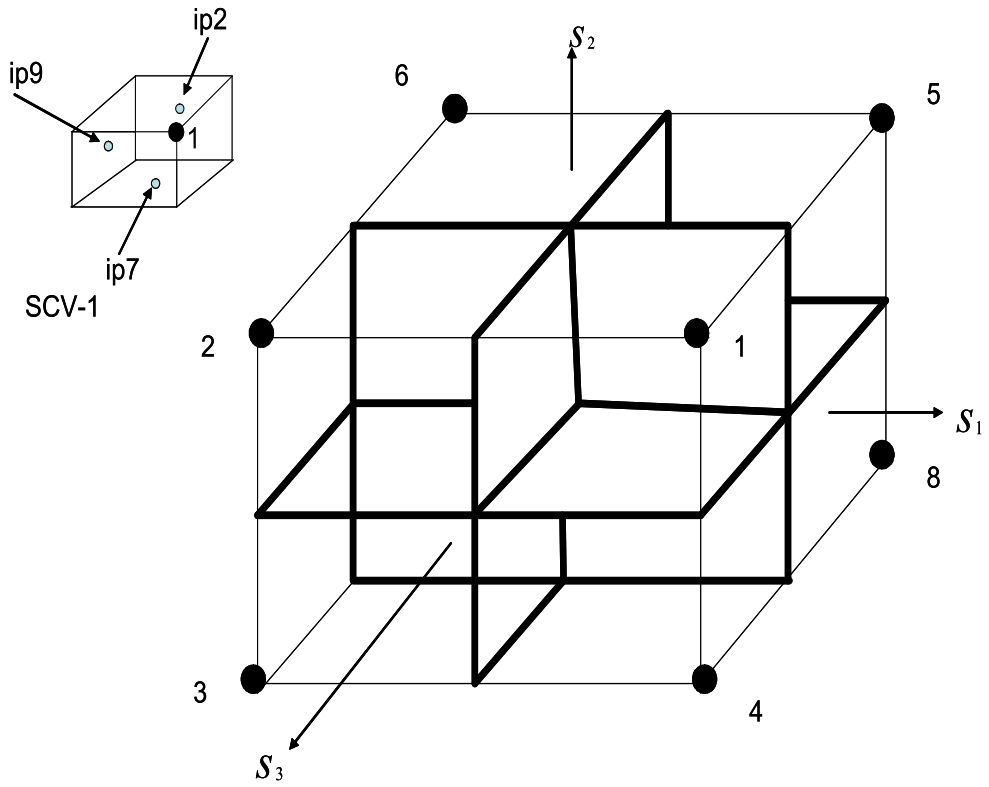


Figure 2.2: The 3–dimensional hexagonal element

$$N_3 = \frac{1}{8}(1 - s_1)(1 - s_2)(1 + s_3) \tag{2.3.7}$$

$$N_4 = \frac{1}{8}(1 + s_1)(1 - s_2)(1 + s_3) \tag{2.3.8}$$

$$N_5 = \frac{1}{8}(1 + s_1)(1 + s_2)(1 - s_3) \tag{2.3.9}$$

$$N_6 = \frac{1}{8}(1 - s_1)(1 + s_2)(1 - s_3) \tag{2.3.10}$$

$$N_7 = \frac{1}{8}(1 - s_1)(1 - s_2)(1 - s_3) \tag{2.3.11}$$

$$N_8 = \frac{1}{8}(1 + s_1)(1 - s_2)(1 - s_3) \tag{2.3.12}$$

From the expression for ϕ in Eqn. (2.3.1), the derivatives can be evaluated according to [58]

$$\frac{\partial \phi}{\partial x} = \sum_{i=1}^{8} \frac{\partial N_i}{\partial x} \phi_i \tag{2.3.13}$$

$$\frac{\partial \phi}{\partial y} = \sum_{i=1}^{8} \frac{\partial N_i}{\partial y} \phi_i \tag{2.3.14}$$

and

$$\frac{\partial \phi}{\partial z} = \sum_{i=1}^{8} \frac{\partial N_i}{\partial z} \phi_i \tag{2.3.15}$$

In terms of local coordinates, the x, y and z derivatives of the shape functions in Eqns. (2.3.13) - (2.3.15) are determined by

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \\ \frac{\partial N_i}{\partial y} \\ \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = \frac{1}{Det(J)} \begin{bmatrix} \left(\frac{\partial y}{\partial s_2}\frac{\partial z}{\partial s_3} - \frac{\partial z}{\partial s_2}\frac{\partial y}{\partial s_3}\right) & \left(\frac{\partial z}{\partial s_1}\frac{\partial y}{\partial s_3} - \frac{\partial y}{\partial s_1}\frac{\partial z}{\partial s_3}\right) & \left(\frac{\partial y}{\partial s_1}\frac{\partial z}{\partial s_2} - \frac{\partial z}{\partial s_1}\frac{\partial y}{\partial s_2}\right) \\ \\ \left(\frac{\partial z}{\partial s_2}\frac{\partial x}{\partial s_3} - \frac{\partial x}{\partial s_2}\frac{\partial z}{\partial s_3}\right) & \left(\frac{\partial x}{\partial s_1}\frac{\partial z}{\partial s_3} - \frac{\partial z}{\partial s_1}\frac{\partial x}{\partial s_3}\right) & \left(\frac{\partial z}{\partial s_1}\frac{\partial x}{\partial s_2} - \frac{\partial x}{\partial s_1}\frac{\partial z}{\partial s_2}\right) \\ \\ \left(\frac{\partial x}{\partial s_2}\frac{\partial y}{\partial s_3} - \frac{\partial y}{\partial s_2}\frac{\partial x}{\partial s_3}\right) & \left(\frac{\partial y}{\partial s_1}\frac{\partial x}{\partial s_3} - \frac{\partial x}{\partial s_1}\frac{\partial y}{\partial s_3}\right) & \left(\frac{\partial x}{\partial s_1}\frac{\partial y}{\partial s_2} - \frac{\partial y}{\partial s_1}\frac{\partial x}{\partial s_2}\right) \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial s_1} \\ \\ \frac{\partial N_i}{\partial s_2} \\ \\ \frac{\partial N_i}{\partial s_3} \end{Bmatrix} \tag{2.3.16}$$

where the Jacobian determinant is

$$Det(J) = \frac{\partial x}{\partial s_1}\left(\frac{\partial y}{\partial s_2}\frac{\partial z}{\partial s_3} - \frac{\partial z}{\partial s_2}\frac{\partial y}{\partial s_3}\right) - \frac{\partial y}{\partial s_1}\left(\frac{\partial x}{\partial s_2}\frac{\partial z}{\partial s_3} - \frac{\partial z}{\partial s_2}\frac{\partial x}{\partial s_3}\right) + \frac{\partial z}{\partial s_1}\left(\frac{\partial x}{\partial s_2}\frac{\partial y}{\partial s_3} - \frac{\partial y}{\partial s_2}\frac{\partial x}{\partial s_3}\right) \tag{2.3.17}$$

Considering the general area segment of Fig. 2.3 being traversed in the direction 1-2-3, the outward normal, $d\vec{S}$, as shown in the figure can be expressed as

$$d\vec{S} = \Delta y \Delta z \; \hat{i} + \Delta x \Delta z \; \hat{j} + \Delta x \Delta y \; \hat{k} \tag{2.3.18}$$

This result will be valid irrespective of the orientation of the area segment, provided that the signs for each discretized differential in Eqn. (2.3.18) correspond to a consistent direction of traversal of the area segment. The elemental volume is given by [59]

$$dV = |Det(J)| \Delta s_1 \; \Delta s_2 \; \Delta s_3 \tag{2.3.19}$$

A hybrid finite element/volume formulation offers a unique advantage over other formulations, as it permits a direct, physical interpretation of the discrete algebraic equations. It enforces conservation constraints for finite control volumes distributed throughout the domain. Also, it permits complete geometric flexibility of finite elements. In Fig. 2.2, the $s_1 = 0$, $s_2 = 0$, and $s_3 = 0$ planes divide the finite–element into 8 sub–control volumes, each of which is associated with a node. For example, the sub–control volume corresponding to node 1 ($SCV - 1$) is the region bounded by $0 \leq s_1 \leq 1$, $0 \leq s_2 \leq 1$, and $0 \leq s_3 \leq 1$.

A complete control volume for a specific node is formed with 8 sub–control volumes from the 8 surrounding finite elements joined to that node (see Fig. 2.4). Each control volume is uniquely associated with a single node (see large dot in Fig. 2.4) common to all of the 8 finite elements. Each finite element contributes 3 faces to the edges of the complete control volume.

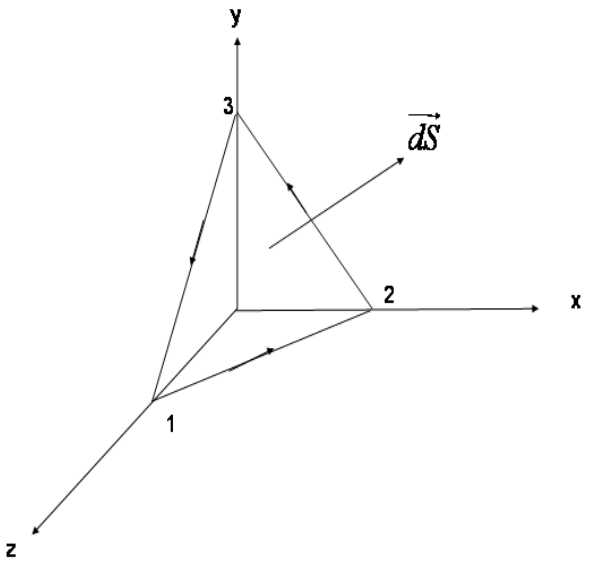


Figure 2.3: Flux vector on a traversed plane

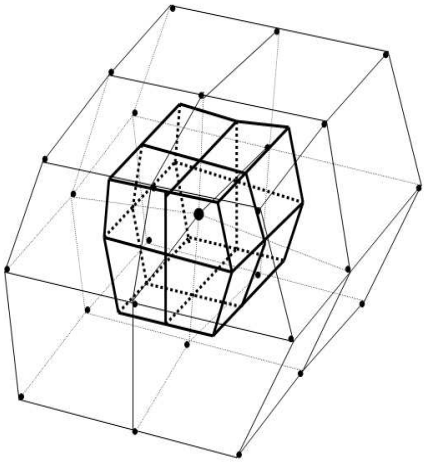


Figure 2.4: Effective control volume in CVFEM

2.4 Discretization of Scalar Transport Equation

The conservation principle for ϕ may be applied to a finite control volume, V, fixed in space and encompassed by a surface, S. Integrating Eqn. (2.2.1) over a control volume and applying the Gauss Theorem,

$$\int_V \frac{\partial(\rho\phi)}{\partial t} V + \int_S (\rho \mathbf{v}\phi) \cdot \vec{dn} + \int_S (\Gamma \nabla\phi) \cdot \vec{dn} = \int_V \hat{S} dV \tag{2.4.1}$$

where $\vec{n}$ is the unit outward normal to the surface, S.

The surface flux terms of Eqn. (2.4.1) will be approximated by each flux evaluated at the surface mid–point, multiplied by the area of that surface. All calculations are performed at an element level, thereby independently of the overall mesh configuration [4, 58]. The locations of the mid-points, known as the integration points, are denoted by $ip1$ through $ip12$, as shown in Fig. 2.5.

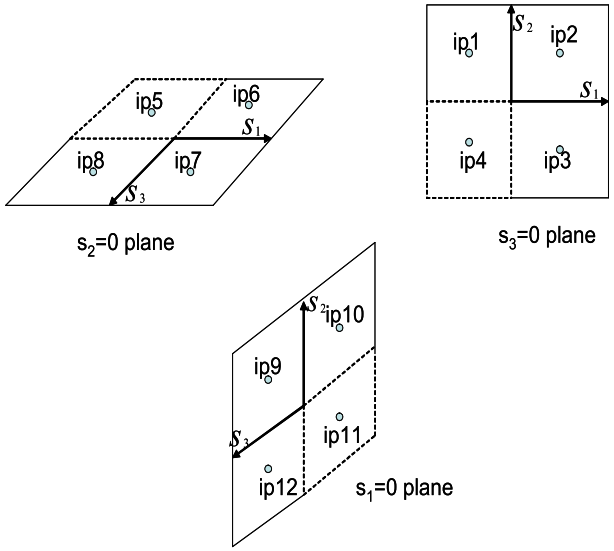


Figure 2.5: The planes of integration points

The equation governing the conservation of the scalar, ϕ, for a discrete volume can be written for sub–control volume 1 using the nomenclature in Fig. 2.2 and Fig. 2.5, in the following form,

$$Q_{ip2} + Q_{ip7} + Q_{ip9} + \int_{SCV1} \hat{S}\, dV = \frac{\partial}{\partial t} \int_{SCV1} \rho\phi\, dV \tag{2.4.2}$$

Note that the flows of ϕ out/into the sub–control volume that are exterior to the element, as contributions from either of the adjacent elements or the domain boundary, are not included in Eqn. (2.4.2). These flows do not require consideration at an element level, since their contributions are mutually self–cancelled from interior element surfaces through the assembly procedure, or through the application of the closure procedure at exterior element surfaces when boundary conditions are applied.

The first three terms in Eqn. (2.4.2) constitute the flows within the element sub–control volume. They contain both diffusive, Q^d, and advective, Q^c, components, i.e.,

$$Q_{ip2} = Q^d_{ip2} + Q^c_{ip2} \tag{2.4.3}$$

The diffusive component, Q^d_{ip2}, for example, is given by

$$Q^d_{ip2} = -\int_S \mathbf{q}^d \cdot \, d\vec{S} \tag{2.4.4}$$

where the diffusive flux, $\mathbf{q}^d$, is given by

$$\mathbf{q}^d = -\Gamma_x \frac{\partial \phi}{\partial x}\hat{i} - \Gamma_y \frac{\partial \phi}{\partial y}\hat{j} - \Gamma_z \frac{\partial \phi}{\partial z}\hat{k} \tag{2.4.5}$$

Using the mid–point approximation for the integration in Eqn. (2.4.4), the result is obtained that

$$Q^d_{ip2} = \sum_{j=1}^{8} \left(\Gamma_x \left[\frac{\partial N_j}{\partial x}\right]^{ip2} \Delta y \Delta z + \Gamma_y \left[\frac{\partial N_j}{\partial y}\right]^{ip2} \Delta x \Delta z + \Gamma_z \left[\frac{\partial N_j}{\partial z}\right]^{ip2} \Delta x \Delta y \right) \phi_j \tag{2.4.6}$$

The advective component, Q^c_{ip2}, for this example is written as

$$Q^c_{ip2} = -\int_S \mathbf{q}^c \cdot \, d\vec{S} \tag{2.4.7}$$

where the advective flux, $\mathbf{q}^c$, is given by

$$\mathbf{q}^c = \rho u \phi \hat{i} + \rho v \phi \hat{j} + \rho w \phi \hat{k} \tag{2.4.8}$$

Using the mid–point approximation,

$$Q^c_{ip2} = -\rho u_{ip2} \, \Delta y \Delta z \, \phi_{ip2} - \rho v_{ip2} \, \Delta x \Delta z \, \phi_{ip2} - \rho w_{ip2} \, \Delta x \Delta y \, \phi_{ip2} \tag{2.4.9}$$

The elemental distances, Δx, Δy, and Δz in Eqns. (2.4.6) and (2.4.9) are determined based on the local derivatives of shape functions in Eqns. (2.3.5)-(2.3.12), i.e.,

$$\Delta x = \left(\sum_{j=1}^{8} \frac{\partial N_j}{\partial s_1} x_j \right) \Delta s_1 + \left(\sum_{j=1}^{8} \frac{\partial N_j}{\partial s_2} x_j \right) \Delta s_2 + \left(\sum_{j=1}^{8} \frac{\partial N_j}{\partial s_3} x_j \right) \Delta s_3 \qquad (2.4.10)$$

where each Δs refers to the difference of the local coordinate across the sub–surface. Similar calculations are performed for Δy and Δz.

Upwind coefficients are needed in Eqn. (2.4.9) when relating the integration point values to nodal values. These coefficients will be denoted by $[N_{up}]$, and their detailed evaluation will be described in the next chapter. Then, Eqn. (2.4.9) becomes

$$Q^c_{ip2} = \sum_{j=1}^{8} \left(-\rho u_{ip2} \Delta y \Delta z - \rho v_{ip2} \Delta x \Delta z - \rho w_{ip2} \Delta x \Delta y \right) [N_{up}]_j \phi_j \qquad (2.4.11)$$

Similar calculations are performed for the other advective fluxes contributing to the $(SCV-1)$ stiffness equation, namely Q_{ip7} and Q_{ip9}.

The combined diffusive and advective fluxes at $ip2$ can be written as

$$Q^{d,c}_{ip2} = \sum_{j=1}^{8} \left(Q^d_{ip2,j} + Q^c_{ip2,j} \right) \phi_j \qquad (2.4.12)$$

When added to the other sub–surface fluxes, namely $Q^{d,c}_{ip7}$ and $Q^{d,c}_{ip9}$, the contributions to $(SCV-1)$ are completed for that element. The subscript j refers to the node of influence, while another subscript, i, is added to accommodate each sub–control volume of the element $(i = 1, 2, ..., 8)$.

The transient term in Eqn. (2.4.2) is approximated by

$$\frac{\partial}{\partial t} \int_{SCV1} \rho\phi \, dV = \beta_1 (\phi_1^{n+1} - \phi_1^n) \qquad (2.4.13)$$

where

$$\beta_1 \;=\; \frac{\rho}{\Delta t}|Det(J_1)|$$

and the superscripts $n+1$ and n refer to current and previous time levels, respectively. The remaining source term in Eqn. (2.4.2) is integrated spatially over $SCV-1$, based on its given (or known) spatial distribution.

The element property equations can now be formed by adding together all of the previously described terms into Eqn. (2.4.2), so that

$$\left([\beta]_{i,j} - [Q^{d,c}]_{i,j}\right) \{\phi\}_j^{n+1} = (\hat{S})_i + [\beta]_{i,j}\{\phi\}_j^n \tag{2.4.14}$$

where $i = 1, 2, ..., 8$, and $[\beta]_{i,j}$ is a diagonal matrix with entries of $\rho|Det(J_i)|/\Delta t$ on the diagonal of row i. The bracketed term on the left side represents the elemental stiffness matrix. The global stiffness matrix is obtained after all elemental matrices are assembled to establish the complete control volume equation.

When the advective fluxes were constructed, the upwind coefficients, $[N_{up}]_j$, were needed to express the integration point variables in terms of nodal variables. In the next chapter, a new upwind method (called NISUS; Non–Inverted Skew Upwind Scheme) will be outlined for a 3–D control volume-based finite element method (CVFEM).

Chapter 3

Finite Element Formulation with Mass Weighting Convective Upwinding

In this chapter, a new convective upwinding procedure is described for discretization of the non–linear convective term in the transport equations. These convective terms were introduced in the previous chapter and their detail formulation will be developed in this chapter. Mass flow weighting of the convective flux across the sub-control volumes will be used as a basis for the upwinding procedure. Three variations of the skew-upwind scheme are developed, with two schemes applicable to the FEM in this chapter. The third method, which is applicable to an FVM, will be described in the next chapter.

3.1 Mass-weighted Upwinding

When evaluating the convective term at integration points in Eqn. (2.4.1), past skew upwind schemes have expressed the integration point variables implicitly in terms of other integration point and nodal variables. For example, the positive coefficient

scheme [4] applies 2–D influences from both integration points and nodal points within the complete element convective analysis. However, extending this method to 3–D problems will have inter–relationships leading to elemental influence coefficient matrices involving 12 integration point and 8 nodal variables for an 8-noded hexahedral element. It implies that matrix inversion will be needed to explicitly represent the integration point variables in terms of nodal variables. This has considerable impact on CPU run–time, particularly for 3–D problems. As a result, an alternative non-inverted approach will be developed in this chapter. Avoiding such inversion, while maintaining or improving accuracy, is a worthwhile objective, particularly for reducing CPU run–time in 3–D simulations. This chapter develops and compares two alternative non-inverted schemes, based on asymmetric (3–node / 3–point) and symmetric (4–node / 8–point) mass–weighted interpolation. An extended variant of NISUS is formulated for a SIMPLEC staggered grid formulation in the next chapter.

While the components of velocity at the integration point in Eqn. (2.4.9) are determined using tri–linear shape functions, the approximation of the convected scalar value requires determination of the upwind value. The novelty of the newly proposed scheme lies in the possibility of obtaining this upwind value without weighting of the nearest integration points on the upstream value. Upwind interpolation generally requires the choice between the following three possibilities: (1) direct nodal points, (2) nodal and integration points, or (3) integration points, in the determination of the upwind value. The proposed schemes are developed based on the first type and effective use of the conservation of mass principle, at the sub–control surface on the upstream side of the integration point. Since the mass flow entering each sub–control volume is used in the weighting factor on the node associated with the sub–control

volume, this nodal value shares the contribution at the adjoining two integration points according to the mass-flow rate at each face.

3.2 3-Node / 3-Point Non-Inverted Skew Upwind Scheme

For an effective control volume shown in Fig. 3.1, 8 sub–control volumes with 12 integration points are involved. Consider $ip9$ in the first sub–control volume. The direction of the flow through this point is required, in order to establish the upstream side of the sub–control face. The 3–Node / 3–Point Non-Inverted Skew Upwind Scheme (NISUS) first evaluates the sub-surface mass flows, i.e., $\dot{m}_1$, $\dot{m}_2$, $\dot{m}_7$, $\dot{m}_8$, and $\dot{m}_9$. The mass flows entering each sub–control volume through the external faces, $\dot{M}_1$ and $\dot{M}_2$, are obtained as

$$\dot{M}_1 = \dot{m}_2 + \dot{m}_7 + \dot{m}_9 \qquad \text{,for } \dot{m}_9 \leq 0 \tag{3.2.1}$$

$$\dot{M}_2 = \dot{m}_1 + \dot{m}_8 - \dot{m}_9 \qquad \text{,for } \dot{m}_9 > 0 \tag{3.2.2}$$

The net mass fluxes through the two neighbouring upstream integration points constitute the remaining upwind influences. For instance, considering the current upwind effect, $\dot{m}_1$ and $\dot{m}_8$ will be evaluated if $\dot{m}_9 > 0$; or $\dot{m}_2$ and $\dot{m}_7$ will be evaluated if $\dot{m}_9 \leq 0$. The ratio of each of these influences over the total influence constitutes the coefficients of the desired upwind interpolation function. Therefore, by evaluating the sign of flux through $ip9$ (see Fig. 2.5), the upwind value, ϕ_{up}, is evaluated as

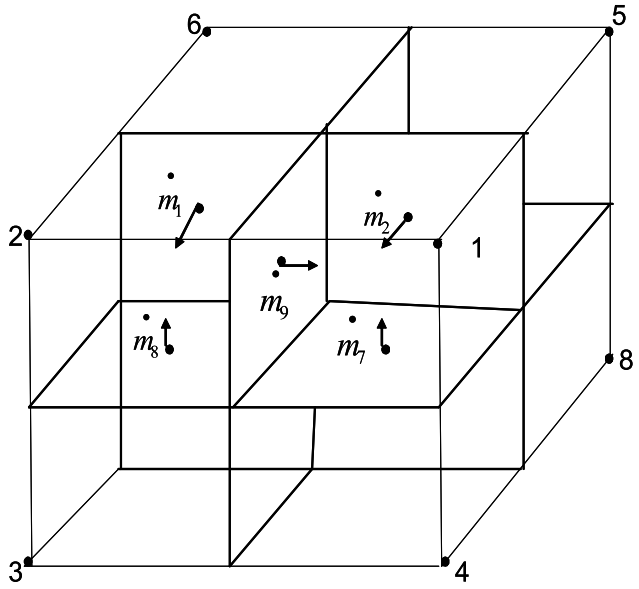


Figure 3.1: Flux across each sub–surface with 3–node / 3–point NISUS

$$\phi_{up} = \begin{cases} \frac{\lfloor \dot{m}_8 \rfloor \, \Phi_3 + \lfloor \dot{m}_1 \rfloor \, \Phi_6 + \lfloor -\dot{M}_2 \rfloor \, \Phi_2}{\lfloor \dot{m}_8 \rfloor + \lfloor \dot{m}_1 \rfloor + \lfloor -\dot{M}_2 \rfloor} & , \quad \text{for } \dot{m}_9 > 0 \\ \\ \frac{\lfloor \dot{m}_2 \rfloor \, \phi_5 + \lfloor \dot{m}_7 \rfloor \, \Phi_4 + \lfloor \dot{M}_1 \rfloor \, \Phi_1}{\lfloor \dot{m}_2 \rfloor + \lfloor \dot{m}_7 \rfloor + \lfloor \dot{M}_1 \rfloor} & , \quad \text{for } \dot{m}_9 \leq 0 \end{cases} \tag{3.2.3}$$

where the positive–definite $\lfloor . \rfloor$ operator is defined as

$$\lfloor x \rfloor = \begin{cases} x & , \quad if \quad x \geq 0 \\ 0 & , \quad if \quad x < 0 \end{cases} \tag{3.2.4}$$

Equation (3.2.3) can be written as

$$\phi_{up} = \sum_{i=1}^{8} N_{up,i} \phi_i \tag{3.2.5}$$

where N_{up} is the upwind influence function and ϕ_i are the nodal values. The magnitude of the upwind scalar value at $ip9$ is thereby approximated. The same procedure

is applied for all 12 integration points. These values are required, in order to complete the approximation of the integration point value (section 3.4).

3.3 4-Node / 8-Point Non-Inverted Skew Upwind Scheme

With eight points under consideration (including the integration point), a symmetric mass weighting uses a 4–node / 8–point representation for upwind interpolation. By increasing the nodal influence and symmetry of upwind coefficients, additional accuracy and stability of the convection modeling may be realized. For instance, considering the integration point denoted by $ip9$ (see Figs. 2.5 and 3.2), the effective mass–flow weighting on the node associated with each upstream sub–control volume is evaluated as follows,

$$\dot{M}_1 = \dot{m}_1 + \dot{m}_9 - \dot{m}_4 \tag{3.3.1}$$

$$\dot{M}_4 = \dot{m}_3 + \dot{m}_4 - \dot{m}_{12} \tag{3.3.2}$$

$$\dot{M}_5 = \dot{m}_5 - \dot{m}_8 - \dot{m}_9 \tag{3.3.3}$$

$$\dot{M}_8 = \dot{m}_7 + \dot{m}_8 - \dot{m}_{12} \tag{3.3.4}$$

while the total influence is

$$\lfloor \dot{M}_{TOT} \rfloor = \lfloor \dot{M}_1 \rfloor + \lfloor \dot{M}_4 \rfloor + \lfloor \dot{M}_5 \rfloor + \lfloor \dot{M}_8 \rfloor \tag{3.3.5}$$

The ratio of each of these mass–flow influences over the total influence constitutes the coefficients of the desired upwind interpolation function. The upwind interpolation function, ϕ_{up}, becomes

$$\phi_{up} = \frac{\lfloor \dot{M}_1 \rfloor}{\lfloor \dot{M}_{TOT} \rfloor}\Phi_1 + \frac{\lfloor \dot{M}_4 \rfloor}{\lfloor \dot{M}_{TOT} \rfloor}\Phi_4 + \frac{\lfloor \dot{M}_5 \rfloor}{\lfloor \dot{M}_{TOT} \rfloor}\Phi_5 + \frac{\lfloor \dot{M}_8 \rfloor}{\lfloor \dot{M}_{TOT} \rfloor}\Phi_8 \tag{3.3.6}$$

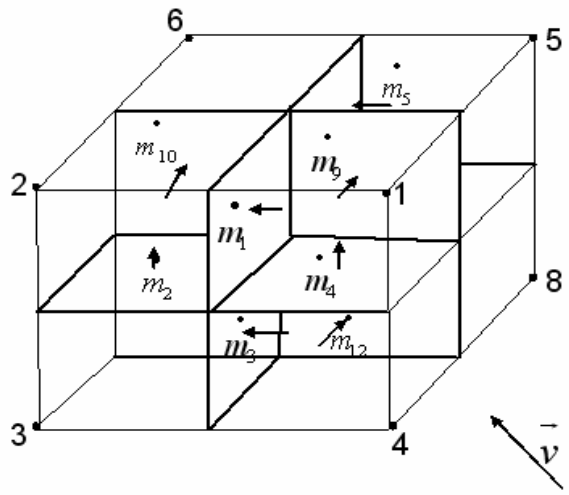


Figure 3.2: Flux across each sub–surfaces with 4-node / 8-point NISUS

where the positive–definite operator, $\lfloor . \rfloor$, defined in Eqn. (3.2.4) is maintained. Equations (3.2.3) and (3.3.6), together with analogous results for other upwind points, give results for upwind variables explicitly in terms of nodal variables. Based on the coefficients pre–multiplying each nodal variable, the upwind variable (i.e., Eqn. 3.3.6) can be expressed as follows,

$$\phi_{up} = \sum_{i=1}^{8} N_{up,i} \phi_i \tag{3.3.7}$$

where N_{up} represents the upwind influence coefficient and i refers to nodal values of the dependent scalar.

Similarly, the x_{up}, y_{up}, and z_{up} positions of the upwind point are evaluated as

$$x_{up} = \sum_{i=1}^{8} N_{up,i} x_i \tag{3.3.8}$$

$$y_{up} = \sum_{i=1}^{8} N_{up,i} y_i \tag{3.3.9}$$

$$z_{up} = \sum_{i=1}^{8} N_{up,i} z_i \tag{3.3.10}$$

Several important characteristics of the upwind influence coefficients, $N_{up,i}$, can be observed from Eqns. (3.3.7)-(3.3.10). Firstly, their values are each less than unity, as required for such interpolation in Eqns. (3.3.8)-(3.3.10). Secondly, the values are all positive, and their sum over $i = 1, 2, ..., 8$ is unity, as required. Thus, they can be considered similarly as the interpolation functions in Eqns. (2.3.5)-(2.3.12), except those previous interpolations are spatially tri–linear, whereas $N_{up,i}$ reflects a mass-weighted (non–linear) interpolation. Unlike previous skew upwind interpolations to the upstream edge of an element [3, 4], the current method may produce higher accuracy since the upwind point may be closer to the integration point (not necessarily at the edge of the element). With this closer proximity, the upwind value may better reflect the upstream convective differencing. Furthermore, upwinding to nearby integration points is not needed, so local inversion of the upwind coefficient matrix is avoided.

The convective length scale represents the distance between the integration point, ip, and the upwind point, up, and it is evaluated as

$$C_{up} = \sqrt{(x_{up} - x_{ip})^2 + (y_{up} - y_{ip})^2 + (z_{up} - z_{ip})^2} \tag{3.3.11}$$

The integration point variable in the convective term of Eqn. (2.4.1) is influenced locally by both advective and diffusive processes. In the following section, the previous upwind values will be incorporated into the integration point equation involving these local transport processes.

3.4 Assembly of Integration Point Variables

In order to obtain an expression for the integration point variable in terms of the nodal values, a transport equation involving advection and diffusion is approximated at the integration point, i.e.,

$$\rho\bar{v}\frac{\partial\phi}{\partial\hat{s}} - \Gamma\nabla^2\phi = 0 \tag{3.4.1}$$

where $\bar{v} = \sqrt{u^2 + v^2 + w^2}$ and $\hat{s}$ is the streamline coordinate. The individual terms in Eqn. (3.4.1) are approximated as follows,

$$\rho\bar{v}\frac{\partial\phi}{\partial\hat{s}} \approx \rho\bar{v}\left(\frac{\phi_{ip} - \phi_{up}}{C_{up}}\right) \tag{3.4.2}$$

and

$$\Gamma\nabla^2\phi \approx \Gamma\frac{\left(\sum_{j=1}^{8} N_j|_{ip}\phi_j\right) - \phi_{ip}}{D^2} \tag{3.4.3}$$

where the diffusive length scale, D is adapted from Ref. [60] as

$$\frac{1}{D^2} = \frac{2}{\Delta n^2} + \frac{8}{3\Delta l^2} + \frac{8}{3\Delta m^2} \tag{3.4.4}$$

The normal (Δn) component and the two tangential ($\Delta l, \Delta m$) components represent the three length scales at the integration point. Assembling these approximations into Eqn. (3.4.1),

$$\phi_{ip} = \frac{\rho\bar{v}D^2\sum_{i=1}^{8} N_{up,i}\phi_i + \Gamma C_{up}\sum_{i=1}^{8} N_i\phi_i}{\rho\bar{v}D^2 + \Gamma C_{up}} \tag{3.4.5}$$

It can be observed that Eqn. (3.4.5) gives an explicit representation of ϕ_{ip} in terms of the nodal variables, ϕ_i.

It can be verified that Eqn. (3.4.5) yields the proper elliptic behavior in the diffusive limit ($\bar{v} \longrightarrow 0$), as well as the full upwind influence in the advection limit

($\Gamma \longrightarrow 0$). The result in Eqn. (3.4.5) gives an explicit representation of ϕ_{ip} in terms of the nodal variables. Therefore, local inversion of the full 12×12 integration point influence matrix is avoided. Without this local inversion to express ϕ_{ip} in terms of ϕ_i, computational savings (reduced CPU run-time) can be realized for three–dimensional simulations. Previous schemes have used implicitly related nodal and integration points variables for various reasons, including higher accuracy for re–circulating flows and more robust pressure / velocity coupling to overcome the pressure checkerboard problem [13]. In contrast, NISUS captures the convection influences from the integration point, without requiring local inversion of the coefficient matrices. In chapter 6, it will be shown that NISUS can match or exceed solution accuracy and exhibit faster CPU run–time, as compared with other conventional upwind methods.

3.5 Boundary Condition Implementation

After the assembly of the elemental discrete equations, the influences at the domain boundaries are not yet included for complete conservation of the transport quantity. The boundary flows of ϕ (representing energy, mass or velocity components) out/into the sub-control volume are not included in Eqn. (2.4.2). Hence, the consideration of flow across the boundaries of the domain will be necessary.

With the CVFEM, the boundary control volume (i.e., labelled a-b-c-d-e-f-g-h in Fig. 3.3) for a sample 8-element solution domain, on the edge shown by 1-2-3-4 in Fig. 3.3, is composed of four sub-control volumes from four elements marked 1-4. The boundary integration points are denoted by *aip*, *bip*, *cip*, and *dip*, which represent the integration points associated with the a, b, c, and d sub-boundary control volumes, respectively.

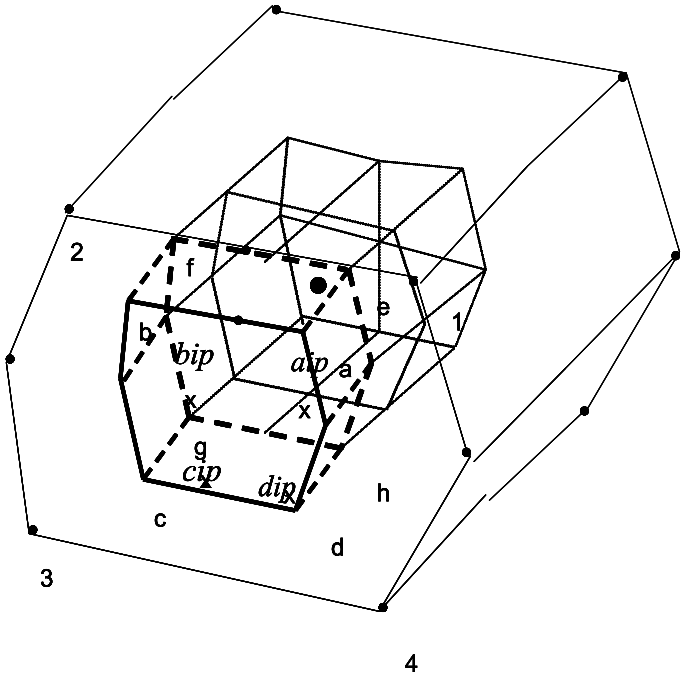


Figure 3.3: Boundary control volume

The convective and diffusive boundary flows can be written as

$$Q_b = Q_b^c + Q_b^d \tag{3.5.1}$$

The complete scalar conservation control–volume equation becomes

$$(\text{Assembled internal equations}) + Q_b = 0 \tag{3.5.2}$$

where Q_b is positive for an inflow across the solution boundary. The convective boundary flow of ϕ is given by

$$Q_b^c = -\int_b \rho\phi\mathbf{v}\cdot dS \tag{3.5.3}$$

Using the mid–point approximation, Eqn. (3.5.3) can be written as

$$Q_b^c \approx \dot{m}_{aip}\phi_{aip} + \dot{m}_{bip}\phi_{bip} + \dot{m}_{cip}\phi_{cip} + \dot{m}_{dip}\phi_{dip} \tag{3.5.4}$$

where $\dot{m}_{aip}$, $\dot{m}_{bip}$, $\dot{m}_{cip}$, and $\dot{m}_{dip}$ are the boundary mass flow rates, positive into the surface associated with a, b, c, and d, respectively. The boundary integration point values, ϕ_{aip}, ϕ_{bip}, ϕ_{cip}, and ϕ_{dip} are expressed in terms of the boundary nodal values, ϕ_a, ϕ_b, ϕ_c, and ϕ_d using the trilinear shape functions for the boundary face.

The diffusive boundary flow is given by

$$Q_b^d = \int_b \mathbf{q}_n dS \tag{3.5.5}$$

Using the mid–point approximation, Eqn. (3.5.5) can be written as

$$Q_b^d = \mathbf{q}_{n1}\Delta S_1 + \mathbf{q}_{n2}\Delta S_2 + \mathbf{q}_{n3}\Delta S_3 + \mathbf{q}_{n4}\Delta S_4 \tag{3.5.6}$$

where $\mathbf{q}_n$ is the normal flux at the boundary integration point, and ΔS is the sub–boundary control surface area. Substituting Eqn. (3.5.4) and Eqn. (3.5.6) into Eqn. (3.5.2), the closed boundary control–volume equations are obtained. In the next chapter, extensions of the NISUS formulation to a staggered grid FVM (finite volume method) will be developed.

Chapter 4

Extended Upwind Scheme in Finite Volume Method with Staggered Grids

4.1 Introduction

Many studies of convective upwinding issues in finite volume methods have been conducted in the past few decades, including pressure induced oscillation [61], false diffusion [62] and slow convergence [13]. In these studies, no ideal method with optimal performance for all problems has been developed. Some of the previous efforts towards addressing the pressure-velocity decoupling have resulted in various algorithms including SIMPLE [13], SIMPLEC [63], and PWIM [64]. A staggered grid method addresses this potential decoupling in finite volume methods. The next section describes the SIMPLEC solution algorithm, while section 4.3 extends NISUS to a SIMPLEC formulation with a staggered grid.

4.2 SIMPLEC Solution Algorithm

Consider incompressible flow governed by conservation of mass and momentum equations, which are generally known as the Navier–Stokes equations. A finite volume

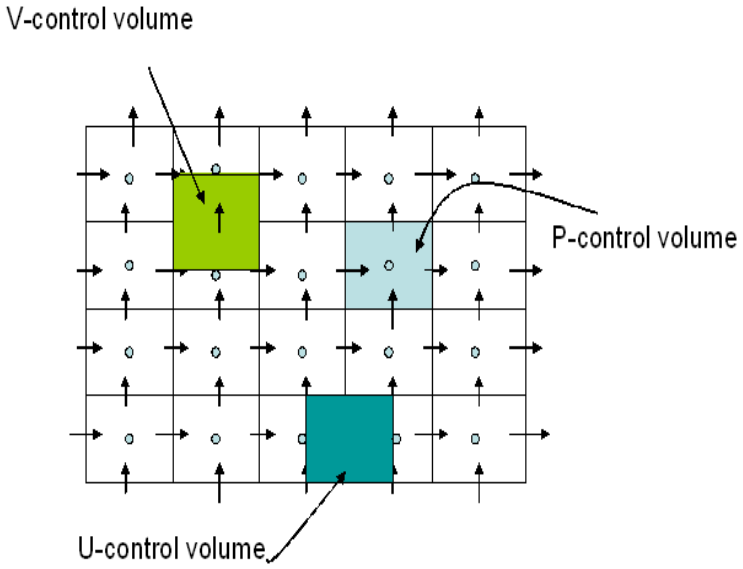


Figure 4.1: Staggered grid arrangement for U, V, and P

method with a staggered grid (see Fig. 4.1) will be considered. Separate U and V control volumes are used for the momentum equations. The pressure is located at the center node in Fig. 4.1, while the velocities are located adjacently (see the U-control volume and V-control volume in Fig. 4.2). The net flows across a control volume boundary are the sums of integrated flux terms in Eqns. (2.2.2)–(2.2.5) for the four control volume edges. This balance can be represented in the well-known summation form as follows [13],

$$a_P \phi_P = \sum a_{nb} \phi_{nb} + b \tag{4.2.1}$$

where a_P, ϕ and b refer to the finite volume coefficient, scalar variable (such as velocity component or temperature) and source term, while the subscripts P and nb refer to central and neighboring nodes, respectively. Eqn. (4.2.1) has been derived and explained fully in Ref. [13]. For example, integration of the mass conservation

equation, Eqn. 2.2.2, over the control volume in Fig. 4.1 yields

$$\rho_e A_e u_e - \rho_w A_w u_w + \rho_n A_n v_n - \rho_s A_s v_s = 0 \tag{4.2.2}$$

Alternatively,

$$a_e^c u_e + a_w^c u_w + a_n^c v_n + a_s^c v_s = 0 \tag{4.2.3}$$

where e, w, s and n represent the east, west, south and north faces of the control volume respectively. The pressure is located at the center node on the domain discretization, while the velocities are stored are moved forward and upward, as indicated by the U-control volume and V-control volume, respectively. By integrating the mass conservation equation (i.e., Eqn. 2.2.2) over the control volume, the algebraic equation in terms of nodal velocities can be obtained [63]. This staggered grid serves to prevent decoupling of pressure and velocity fields, which may lead to a pressure checkerboard [13].

The first step of the SIMPLEC procedure calculates the coefficients relating velocity correction to pressure corrections. The finite-volume equations for U and V in Eqns. (2.2.3) and (2.2.4), respectively, have the form

$$a_p^u u_p = \sum a_{nb}^u u_{nb} + b_p^u + A_p(p_P - p_E) \tag{4.2.4}$$

For the estimated pressure distribution p^*, the resulting $v^*$and u^* fields obtained from the momentum equation satisfy

$$a_p^u u_p^* = \sum a_{nb}^u u_{nb}^* + b_p^u + A_p(p_P^* - p_E^*) \tag{4.2.5}$$

whereas only the correct pressure distribution can satisfy Eqn. (4.2.4). A correction of the pressure by $p' = p - p^*$ is necessary to correct the u^* and v^* fields.

In the SIMPLEC algorithm, a coefficients (d) relating the velocity correction to pressure corrections is evaluated from [65]

$$d = \frac{A_p}{a_p - \sum a^u_{nb} u_{nb}} \tag{4.2.6}$$

When calculating the coefficients for the p' equation, the following steps were followed. The coefficients for the p' equations were calculated using

$$a^p_e = \rho_e A_e d_e \tag{4.2.7}$$

$$a^p_w = \rho_w A_w (d_e)_w \tag{4.2.8}$$

$$a^p_n = \rho_n A_n d_n \tag{4.2.9}$$

$$a^p_s = \rho_s A_s (d_n)_s \tag{4.2.10}$$

Then, the coefficients along the boundaries were modified as follows,

Right

$$a^p_e = 0 \tag{4.2.11}$$

$$a^p_w = \left(a^c_w + a^c_e \frac{a^u_{wf}}{a^u_{pf}} \right) d^{u_w}_e \tag{4.2.12}$$

$$a^p_n = -a^c_n d^{v_p}_n \tag{4.2.13}$$

$$a^p_s = a^c_s d^{v_s}_n \tag{4.2.14}$$

$$a^p_p = a^p_e + a^p_w + a^p_n + a^p_s \tag{4.2.15}$$

Left

$$a^p_w = 0 \tag{4.2.16}$$

$$a^p_e = -\left(a^c_e + a^c_w \frac{a^u_{ef}}{a^u_{pf}} \right) d^{u_e}_e \tag{4.2.17}$$

$$a_n^p = -a_n^c d_n^{v_p} \tag{4.2.18}$$

$$a_s^p = a_s^c d_n^{v_s} \tag{4.2.19}$$

$$a_p^p = a_e^p + a_w^p + a_n^p + a_s^p \tag{4.2.20}$$

Top

$$a_n^p = 0 \tag{4.2.21}$$

$$a_s^p = \left(a_s^c + a_n^c \frac{a_{sf}^v}{a_{pf}^v} \right) d_n^{v_s} \tag{4.2.22}$$

$$a_e^p = -a_e^c d_e^{u_e} \tag{4.2.23}$$

$$a_w^p = a_w^c d_e^{u_w} \tag{4.2.24}$$

$$a_p^p = a_e^p + a_w^p + a_n^p + a_s^p$$

Bottom

$$a_s^p = 0 \tag{4.2.25}$$

$$a_n^p = -\left(a_n^c + a_s^c \frac{a_{nf}^v}{a_{pf}^v} \right) d_n^{v_n} \tag{4.2.26}$$

$$a_e^p = -a_e^c d_e^{u_e} \tag{4.2.27}$$

$$a_w^p = a_w^c d_e^{u_w} \tag{4.2.28}$$

$$a_p^p = a_e^p + a_w^p + a_n^p + a_s^p$$

The coefficients at the fictitious points were set for $p\prime$ equalling unity.

When calculating the source term BPP for the $p\prime$ equation, the source term for the $p\prime$ equation is given by

$$b = (\rho u^* A)_w - (\rho u^* A)_e + (\rho u^* A)_s - (\rho u^* A)_n \tag{4.2.29}$$

The task of the pressure-velocity correction is to ensure that b is zero.

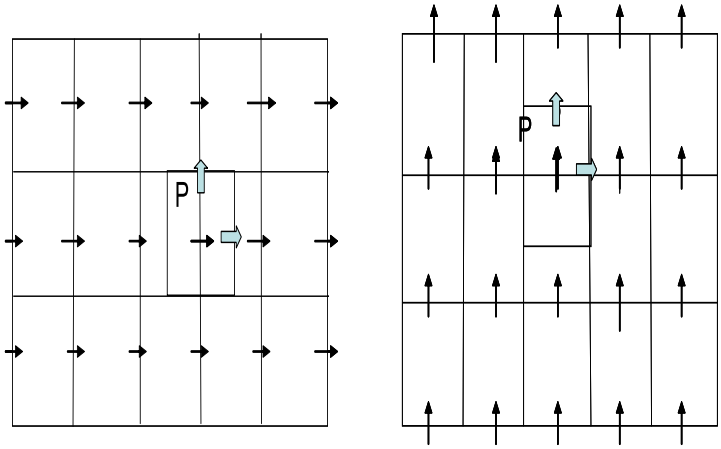


Figure 4.2: U- & V-control volume fluxes at P cell

4.3 NISUS Variant of the Exponential Differencing Scheme

In this section, the NISUS mass weighting approach of convective upwinding (described in previous chapter), will be adapted within a modified Exponential Differencing Scheme (EDS). The conventional EDS approximates the convection variables at the face of the control volume by using only fluxes indicated by the line arrows on Figure 4.2. In the proposed NISUS variant of EDS, flow directionality of mass fluxes is included in the approximation.

In Eqn. (4.2.1), the coefficients a_p and a_{nb} include convective fluxes at the edges of control volumes, which must be determined from interpolation of values in nearby control volumes. These methods of interpolation include UDS, EDS and others. Convective non-linearities generally entail a time-consuming portion of the numerical simulations. UDS uses the nodal value on the upstream side of the control volume, while CDS takes an average value between upstream and downstream nodes [13].

EDS uses a weighting factor to balance advection and diffusion influences based on the local Peclet number. Unlike the conventional EDS, this section adapts NISUS to develop a new upstream weight factor based on mass flow rates and flow directionalities from several adjacent upstream cells. The method expresses coefficients explicitly in terms of nodal values. It does not require local inversion between integration and nodal variables, so it will be called a variant of the Non-Inverted Skew Upwind Scheme (NISUS) with EDS. NISUS will be developed specifically for a staggered grid SIMPLEC formulation.

Within the conventional EDS convective upwinding approach, the approximation of both diffusion (β) and convection (α) weightings is dependent on the local Peclet number. This approach has been shown to have sensitivity to the direction of flow [2]. However, discontinuity of the scalar across the bounding surfaces of tetrahedral elements at high Peclet numbers may occur. In EDS, weighting factors for convection weighting and diffusion weighting are used to predict flux terms leading to the calculation of a_p and a_{nb} coefficients. In particular, the following coefficients can be derived for EDS at the east face [13],

$$\alpha_e = \frac{1}{2} - \frac{\left[exp\left(\frac{1}{2}(Pe)_e\right) - 1\right]}{\left[exp\left((Pe)_e\right) - 1\right]} \tag{4.3.1}$$

$$\beta_e = \frac{(Pe)_e \left[exp\left(\frac{1}{2}(Pe)_e\right) - 1\right]}{\left[exp\left((Pe)_e\right) - 1\right]} \tag{4.3.2}$$

where

$$(Pe)_e = \frac{\rho_e A_e U_e}{\left(\frac{\Gamma_e A_e}{(\delta x)_e}\right)} \tag{4.3.3}$$

The convective weight factor in Eqn. (4.3.1) depends on the local Peclet number. In the calculation of the local Peclet number (Eqn. 4.3.3), the flow directionality

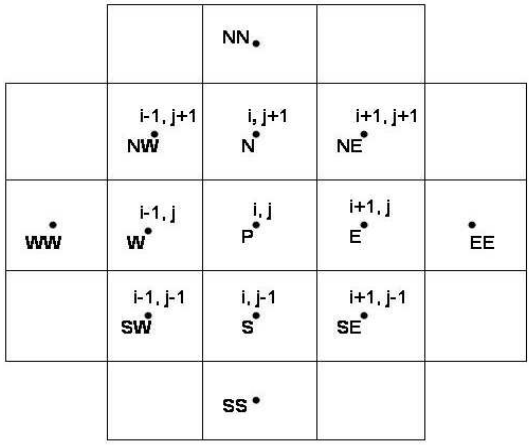


Figure 4.3: Neighboring nodal points to the control volume point P

is not considered. It would be beneficial to accommodate flow directionality in the calculation of α, without jeopardizing the solution convergence and accuracy. Various past methods have been developed for skewed upwinding to accommodate flow directionality [15, 4]. Unlike those past methods, this section accommodates flow directionality through mass flow weighting from multiple upstream control volumes, within the context of a SIMPLEC formulation.

Rather than using the EDS approach of convection weight factors based on the local Peclet number alone, NISUS uses a discrete balance of diffusion and convective flows in the following manner,

$$\dot{m}_e\phi_e - \dot{m}_w\phi_w = \Gamma_e A_e \left(\frac{d\phi}{dx}\right)_e - \Gamma_w A_w \left(\frac{d\phi}{dx}\right)_w \tag{4.3.4}$$

and

$$\dot{m}_n\phi_n - \dot{m}_s\phi_s = \Gamma_n A_n \left(\frac{d\phi}{dx}\right)_n - \Gamma_s A_s \left(\frac{d\phi}{dx}\right)_s \tag{4.3.5}$$

The following interpolations will be used,

$$\phi_e = \alpha_e \phi_P + (1 - \alpha_e)\phi_E \tag{4.3.6}$$

$$\phi_w = \alpha_w \phi_W + (1 - \alpha_w)\phi_P \tag{4.3.7}$$

$$\phi_n = \alpha_n \phi_P + (1 - \alpha_n)\phi_N \tag{4.3.8}$$

$$\phi_s = \alpha_s \phi_S + (1 - \alpha_s)\phi_P \tag{4.3.9}$$

$$\left(\frac{d\phi}{dx}\right)_e = \beta_e \frac{\phi_E - \phi_P}{(\delta x)_e} \tag{4.3.10}$$

$$\left(\frac{d\phi}{dx}\right)_w = \beta_w \frac{\phi_P - \phi_W}{(\delta x)_w} \tag{4.3.11}$$

$$\left(\frac{d\phi}{dy}\right)_n = \beta_n \frac{\phi_N - \phi_P}{(\delta y)_n} \tag{4.3.12}$$

$$\left(\frac{d\phi}{dy}\right)_s = \beta_s \frac{\phi_P - \phi_S}{(\delta y)_s} \tag{4.3.13}$$

In past methods, skewed interpolation of the convective flux to a single upstream cell involves a streamline intersecting the upstream edge to establish weighting between adjacent nodal values [13]. Unlike these past methods, interpolation to multiple upstream cells can be predicted through mass flow rates crossing various cell edges in the upstream direction. Mass flow weighting from multiple upstream cells is a novel approach that will be adopted in this section. With reference to control volume edges depicted in Fig. 4.3, the convection weight factors become

$$\alpha_e = \begin{cases} \frac{\lfloor \dot{m}_{P-E} \rfloor}{\lfloor -\dot{m}_{P-N} \rfloor + \lfloor \dot{m}_{P-W} \rfloor + \lfloor -\dot{m}_{P-S} \rfloor} & , \text{ for } \dot{m}_{P-E} > 0 \\ \frac{\lfloor -\dot{m}_{P-E} \rfloor}{\lfloor -\dot{m}_{E-NE} \rfloor + \lfloor -\dot{m}_{E-EE} \rfloor + \lfloor \dot{m}_{E-SE} \rfloor} & , \text{ for } \dot{m}_{P-E} \leq 0 \end{cases} \tag{4.3.14}$$

$$\alpha_w = \begin{cases} \frac{\lfloor \dot{m}_{P-W} \rfloor}{\lfloor -\dot{m}_{W-NW} \rfloor + \lfloor \dot{m}_{W-WW} \rfloor + \lfloor \dot{m}_{W-SW} \rfloor} & , \text{ for } \dot{m}_{P-W} > 0 \\ \frac{\lfloor -\dot{m}_{P-W} \rfloor}{\lfloor -\dot{m}_{P-N} \rfloor + \lfloor -\dot{m}_{P-E} \rfloor + \lfloor \dot{m}_{P-S} \rfloor} & , \text{ for } \dot{m}_{P-W} \leq 0 \end{cases} \tag{4.3.15}$$

$$\alpha_n = \begin{cases} \frac{\lfloor \dot{m}_{P-N} \rfloor}{\lfloor -\dot{m}_{P-E} \rfloor + \lfloor \dot{m}_{P-W} \rfloor + \lfloor \dot{m}_{P-S} \rfloor} & , \text{ for } \dot{m}_{P-N} > 0 \\ \frac{\lfloor -\dot{m}_{P-N} \rfloor}{\lfloor -\dot{m}_{N-NE} \rfloor + \lfloor -\dot{m}_{N-NN} \rfloor + \lfloor \dot{m}_{N-NW} \rfloor} & , \text{ for } \dot{m}_{P-N} \leq 0 \end{cases} \tag{4.3.16}$$

$$\alpha_s = \begin{cases} \frac{\lfloor \dot{m}_{P-S} \rfloor}{\lfloor -\dot{m}_{S-SE} \rfloor + \lfloor \dot{m}_{S-SW} \rfloor + \lfloor \dot{m}_{S-SS} \rfloor} & , \text{ for } \dot{m}_{P-S} > 0 \\ \frac{\lfloor -\dot{m}_{P-S} \rfloor}{\lfloor -\dot{m}_{P-N} \rfloor + \lfloor -\dot{m}_{P-E} \rfloor + \lfloor \dot{m}_{P-W} \rfloor} & , \text{ for } \dot{m}_{P-S} \leq 0 \end{cases} \tag{4.3.17}$$

The positive–definite operator, $\lfloor . \rfloor$, is defined as

$$\lfloor x \rfloor = \begin{cases} x & , \quad if \quad x \geq 0 \\ 0 & , \quad if \quad x < 0 \end{cases} \tag{4.3.18}$$

From Fig. 4.3, it can be observed that mass weighting extends to multiple upstream cells. For example, the edges P-S and S-SS in Eqn. (4.3.17) refer to the first south edge and second south edge of control volumes below the P control volume in Fig. 4.3. A uni-directional skewed flow is well approximated by an interpolation to a single upstream cell. However, re-circulating flows can be better approximated with skewed upwinding to multiple upstream cells.

The NISUS-Hybrid EDS is a second order differencing scheme with promising potential of better accuracy and stability. Substituting Eqns. (4.3.6) to (4.3.13) into Eqns. (4.3.4) and (4.3.5), the new coefficients for the NISUS-Hybrid EDS becomes

$$\begin{aligned} a_E &= \beta_e D_e + \dot{m}_e \left(1 - \alpha_e\right) \\ a_W &= \beta_w D_w + \dot{m}_w \left(1 - \alpha_w\right) \\ a_N &= \beta_n D_n + \dot{m}_n \left(1 - \alpha_n\right) \\ a_S &= \beta_s D_s + \dot{m}_s \left(1 - \alpha_s\right) \end{aligned} \tag{4.3.19}$$

This completes the NISUS formulation for a staggered grid SIMPLEC method. Before testing the formulation on different example problems (chapter 6), an iterative solver for the resulting algebraic equations will be developed in the next chapter (chapter 5.

Chapter 5

Iterative Solver and Data Storage Format

The development of an iterative solver for the previous numerical schemes will be presented in this chapter. A major challenge in iterative techniques involves the effective manipulation of non–zero entries to minimize data storage. A new data storage format (called CBD; Compressed Banded Data) for a banded coefficient matrix will be developed in this chapter.

5.1 Introduction

Conjugate gradient methods including the Bi–conjugate Gradient Stabilized method (Bi–CGSTAB [66]) and Generalized Minimal Residual method (GMRES(m)) have been widely used for iterative solutions of fluid flow equations. Past studies have successfully applied preconditioned conjugate gradient methods to large-scale systems. For example, a primal / dual methodology was developed by Wolkowicz [67] for solving large sparse systems of equations with a preconditioned conjugate gradient method. An interior point technique avoids difficulties associated with ill–conditioning from

symmetrization of the algebraic equations. A pre–processing step before linearization can reduce the optimality conditions, thereby giving a well–conditioned, over–determined bi–linear operator equation. Furthermore, a cross-over technique near the optimal solution removes a constraint requiring positive–definite iteration values. The method was shown to produce high accuracy in test problems reported by Wolkowicz [67].

Split preconditioning has been shown to improve spectral properties of a matrix system, thereby allowing highly accurate and fast solutions of 2–D elliptic problems (Ali, Evans [68]). The linear system can be derived from minimization of an eigenvalue spectrum associated with the transformed matrix. The preconditioning method was successfully applied within various acceleration iterative methods, including the method of simultaneous displacement, Richardson's method and Chebyshev's method. Conventional preconditioned solvers become impractical for certain physical problems, such as coupled fluid / structure interaction problems. For example, interactions between an acoustic field and elastic structure submerged in a fluid yield a complex sparse matrix structure [69]. A block diagonal preconditioner with a sparse inverse methodology has been developed to accelerate convergence for such coupled systems [69]. Alternatively, Krylov iterative methods with a ILU–Preconditioner [70, 57] can be applied for stability and convergence acceleration.

5.2 Incomplete LU–Preconditioning

A preconditioning technique provides an effective approximation of the coefficient matrix, when solving the linear system of equations iteratively. The development of a

suitable and efficient preconditioner is largely problem dependent. Various preconditioners have been developed previously. A comparison of different preconditioners was presented by Mazzia and Pinii [35]. The incomplete LU decomposition preconditioner is a widely used method, especially when applied to finite element methods [71, 72]. In this chapter, an incomplete LU decomposition is adopted in order to obtain $[M]$, where $[M]$ is the approximated matrix decomposed into a lower square matrix $[L]$ and upper square matrix $[U]$. Both $[L]$ and $[U]$ will be stored in a single–variable compressed data format, called CBD compression.

The IKJ version of ILU decomposition (with no-fill) will be adopted in the present studies. The ILU type preconditioner can be used together with Krylov iterative solvers, including CG [67, 73], CGS [74], Bi-CGSTAB [37, 66], and GMRES [72, 75]. CG (Conjugate Gradient) is an iterative method for large classes of linear systems with symmetric positive definite coefficient and preconditioned matrices. Despite its limitation to symmetric matrices, it is applicable to problems involving hybrid finite element / volume methods (such as a CVFEM). For non-symmetric linear systems, CGS may be implemented instead of CG for problems that are not too large [74, 57]. The GMRES(m) method has been shown to demonstrate good convergence for various practical, large-scale problems [70, 57].

The Bi-conjugate Gradient Stabilized Method (Bi–CGSTAB) was initially developed to solve non-symmetric linear systems of equations. It also exhibits good convergence performance for large-scale systems. In chapter 6 , these variations of the CG method will be tested in several problems involving convective heat transfer. In this chapter, a preconditioning technique based on ILU decomposition is performed after factoring the matrix $[A]$ into a product of $[L]$ (lower triangular matrix) and $[U]$ (upper

triangular matrix). The decomposition of $[A]$ into LU factors is stored in the lower and upper triangular parts of another matrix, $[M]$. As a result, there is no need to explicitly manipulate the coefficients of the product matrix. The two parts are utilized sequentially within the solution routine (based on methods of backward and forward substitution) of an incomplete Cholesky decomposition. An $ILU\,(0)$ decomposition produces an incomplete factorization without any additional non–zero elements beyond the original coefficients of the sparse matrix. For a linear system $(A \cdot x = b)$, the iterative methods in this chapter will be based on projection processes involving Krylov subspaces. These subspaces are spanned by vectors of the form $p\,(A)\,b$, where p is a polynomial. Detailed discussions regarding this associated orthogonalization theory are described in Refs. [41] and [76].

5.3 Compressed Banded Data Format

Efficient storage of the coefficient matrices is an important cost–savings aspect of the computational simulations. Storing coefficient matrices for sparse linear systems can be time-consuming and inefficient. The banded system of equations typically involves $\mathcal{O}\,(n)$ non-zero entries with n unknowns, but a direct storage algorithm entails $\mathcal{O}\,(n^2)$ entries saved in the global matrix. The difference between $\mathcal{O}\,(n)$ and $\mathcal{O}\,(n^2)$ rises rapidly, particularly for three–dimensional problems when the number of grid points increases. Thus, an efficient storage scheme is needed, which takes advantage of the type of matrix generated by the discretization of the governing equations. For example, local stiffness equations are generated independently of the mesh configuration in finite element methods, thereby generating a different global matrix structure than conventional finite difference methods.

In order to reduce storage requirements of large sparse matrices, various methods of data compression have been developed previously. For example, the Compressed Sparse Row (CSR) data structure is a commonly adopted method of data compression. This format consists of three main arrays, namely:

- a real array, $val\,(nz)$, of non–zero elements, a_{ij}, of the matrix, A_{csr}, stored row-by-row from the first row to the last,
- an integer array, $jval\,(nz)$, containing the column pointer of all entries in $val\,(nz)$ and
- an integer array, $ival\,(n+1)$, containing the location pointer of the first non–zero entry on each row of A_{csr} in $val\,(nz)$.

For example, consider the following 6×6 matrix,

$$A_{csr} = \begin{pmatrix} a_{11} & a_{12} & 0 & a_{14} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 & a_{35} & 0 \\ 0 & a_{42} & 0 & a_{44} & a_{45} & 0 \\ 0 & 0 & a_{53} & 0 & a_{55} & a_{56} \\ 0 & 0 & a_{63} & a_{64} & 0 & a_{66} \end{pmatrix} \tag{5.3.1}$$

Table 5.1 outlines how A_{csr} is stored with the CSR format in this example. The storage procedure of A_{csr} with CSR is relatively straightforward, as it involves an unrestrained double-loop sorting of a_{ij} for the non-zero elements. Rather than storing indices for the rows, the CSR structure saves row pointers, with indices pointing to the column and value arrays to give starting positions of each row. In the CSR format, values are stored one row after another. In this way, columns do not need to

be ordered in any particular fashion, while storage is reduced to nnz floating point entries and $(nnz + n + 1)$ integer entries (note: nnz refers to the number of non-zero entries).

Although the row access appears straightforward, data access to particular column entries becomes difficult. For example, transposed matrix / vector multiplications can become inefficient, as certain machine compilers will perform poorly when attempting to recognize that addresses computed in certain column pointers can be re–used when writing that column entry back to memory. An important design criterion for optimal data storage includes ease of data archival and retrieval, as well as minimal memory requirements for a given size of system. Furthermore, the implementation becomes highly complex for banded sparse matrices generated by numerical discretizations with the CVFEM adopted in chapter 2.

For example, consider the compact data storage of CSR with an ILU preconditioned iterative solver [72]. In Ref. [72], a CSR format was adopted for optimal matrix / column vector multiplications in an ILU preconditioning GMRES procedure. The numerical model was presented as follows,

$$[A]\,(x) = (b) \tag{5.3.2}$$

where $[A]$ is the non–symmetric Jacobian matrix $(ndof \times ndof)$, (x) is the finite element solution column vector $(ndof \times 1)$ and (b) is the residual column vector $(ndof \times 1)$. Direct scanning of a sparse banded $(ndof \times band)$ matrix was attempted with CSR, but it can be shown that the compression format cannot be converted with CSR for banded matrices generated by a CVFEM.

Due to CSR's limitations, various other sparse matrix compression algorithms have been developed previously. For example, CSC (Compressed Sparse Column) is

i	$val\,(i)$	$jval\,(i)$	$ival\,(j)$
1	a_{11}	1	1
2	a_{12}	2	4
3	a_{14}	4	7
4	a_{21}	1	10
5	a_{22}	2	13
6	a_{23}	3	16
7	a_{32}	2	19
8	a_{33}	3	
9	a_{35}	5	
10	a_{42}	2	
11	a_{44}	4	
12	a_{45}	5	
13	a_{53}	3	
14	a_{55}	5	
15	a_{56}	6	
16	a_{63}	3	
17	a_{64}	4	
18	a_{66}	6	

Table 5.1: Example of CSR Storage (note: column pointer = jval(i), while val(i) = matrix entry)

similar to CSR, except that it stores row indices and column pointers (rather than row pointers). In block CSR, there is an underlying mesh with several unknowns associated with each grid point. A small dense matrix is stored for each mesh point. MSR (Modified Sparse Row) is similar to CSR, except that the diagonal matrix entries are stored in the first set of positions within the storage array. Other methods include skyline storage, jagged diagonal storage, and so forth. Commercial software is available to translate back and forth between various types of sparse matrix data structures.

However, these methods do not address unique features of banded sparse matrices generated by a CVFEM in chapter 2. Unlike these past methods, a new CBD (Compressed Banded Data) format will be developed, in order to take advantage of the unique structure of matrices generated from hybrid finite volume / element methods, particularly a CVFEM. Unlike the previous matrix, A_{csr}, the assembly procedure of the CVFEM stores all entries of the local stiffness matrix in a banded structure within the global stiffness matrix. Inter-connectivity rules between local nodes and global nodes are used to establish rows / columns of the global matrix where coefficients are stored. For example, the equivalent banded matrix (corresponding to A_{csr} in the previous example) is saved as

$$A_{cbd} = \begin{pmatrix} 0 & 0 & 0 & a_{11} & a_{12} & 0 & a_{14} \\ 0 & 0 & a_{21} & a_{22} & a_{23} & 0 & 0 \\ 0 & 0 & a_{32} & a_{33} & 0 & a_{35} & 0 \\ 0 & a_{42} & 0 & a_{44} & a_{45} & 0 & 0 \\ 0 & a_{53} & 0 & a_{55} & a_{56} & 0 & 0 \\ a_{63} & a_{64} & 0 & a_{66} & 0 & 0 & 0 \end{pmatrix} \tag{5.3.3}$$

where $(ndof \times nband) = 6 \times 7$. It can be observed that special consideration is needed with respect to the upper left and lower right corners of the original square

matrix (A_{csr}). Zero entries are added to the left and right sides of diagonal entries in these corners, in order to conform within the banded format. The iterative solver must recognize that the diagonal entries of a square matrix are stored along a vertical column within the banded matrix, midway within the matrix. Thus, additional parameters such as the semi-bandwidth are needed to retrieve coefficients from the banded matrix. Consider the following pseudo-code for the new data compression algorithm that stores data in a CBD format. This algorithm allows direct scanning and storage of non-zero entries in a banded matrix, without changing the internal coding of a CVFEM.

for $i \longleftarrow 1$ to n

for $j \longleftarrow 1$ to n and if $(i,j) \notin P$

$t \longleftarrow (j - i) + \text{semi}$

$val\,(nz) \longleftarrow a_{it}$

endfor

endfor

In this algorithm, nz refers to an index of the non-zero element and P is the zero pattern set defined by

$$P \subset [(i,j) \mid i \neq j; 1 \leq i,j \leq n] \tag{5.3.4}$$

This implementation of data storage involves a double overloading of the scanning procedure for accurate dynamic allocation of the three linear arrays (see Table 5.2 for CBD storage of A_{csr}). In the CBD format, the dimension of the data variable (val) and the column pointer ($jval$) is given by the number of non-zeros in the coefficient matrix (nz), while the dimension of the position pointer to the first non-zero entry

val	a_{11}	a_{12}	a_{14}	a_{21}	a_{22}	a_{23}	...	a_{56}	a_{63}	a_{64}	a_{66}
$jval$	1	2	4	1	2	3	...	6	3	4	6
$ival$	1	4	7	10	13	16	19				

Table 5.2: Format of Memory Allocation in Compressed Banded Data Storage

in each row ($ival$) is $(ndof + 1)$.

It is worthwhile to compare the memory requirements of the CBD format with direct scanning of the same coefficient matrix without CBD data compression. The storage requirement for a direct solver in megabytes becomes [72]

$$\text{Storage} = [(band + 1) \cdot ndof + 0.5] \left(\frac{8}{2^{20}} \right) \tag{5.3.5}$$

In contrast, the amount of memory required with the CBD format is significantly reduced, i.e.,

$$\text{Storage} = \left[\frac{3\,(nnz)}{2} + ndof + 0.5 \right] \left(\frac{8}{2^{20}} \right) \tag{5.3.6}$$

where nnz refers to the number of non–zero elements. In addition to reduced storage requirements, it will be shown in chapter 6 that the CBD structure has significance in reducing CPU run–time for large–scale systems.

5.4 Tri-Diagonal Matrix Algorithm

The Tri-Diagonal Matrix Algorithm (TDMA) is also known as the Thomas Algorithm. This is a specialized algorithm for obtaining the solution to a set of equations with a tri-diagonal coefficient matrix. This algorithm is developed and used for the NISUS variant EDS developed in section 4.3. A summary of the steps is listed as follows.

Carry out a forward substitution to eliminate the lower diagonal in the matrix.

This entails recursive relations in terms of P_i and Q_i, obtained through the relationship of ϕ involving P_i and Q_i as coefficients. With a tri-diagonal matrix system of equations for the field variable ϕ,

$$(a_P)_i \phi_i = (a_W)_i \phi_{i-1} + (a_E)_i \phi_{i+1} + (b_P)_i \tag{5.4.1}$$

The following relation is used,

$$\phi_i = P_i \phi_{i+1} + Q_i \tag{5.4.2}$$

From a recursive analysis, the following relations are obtained:

$$P_i = \frac{(a_E)_i}{((a_P)_i - (a_W)_i P_{i-1})} \tag{5.4.3}$$

and

$$Q_i = \frac{(a_W)_i Q_{i-1} + (b_P)_i}{((a_P)_i - (a_W)_i P_{i-1})} \tag{5.4.4}$$

Then, calculate ϕ_N, where N is the number of equations. Using ϕ_N, then proceed with back substitution to find the remaining solution vector values. A line Gauss-Seidel method is utilized for a linear directional sweep in the solution procedure. For a two-dimensional solver, a South-to-North sweep and then West-to-East sweep orders, is applied in sequence. In the next chapter, numerical results based on the aforementioned solver and procedures will be presented.

Chapter 6

Heat Transfer and Fluid Flow Results

In this chapter, heat transfer and fluid flow problems are studied. The results from two NISUS schemes formulated within the CVFEM and various Krylov iterative techniques with the new Compressed Banded Data format (CBD) are presented. The results are validated and compared with other schemes, in order to assess the performance of the new methods.

6.1 Introduction

The performance of past convective schemes has often been subjectively based on compromise [5]. Major factors affecting the performance of an advection scheme are summarized below.

- The size and configuration of the solution domain affect performance. A smaller domain leaves less flexibility in the type of element(s) used for the problem discretization.

- The type of flow within the domain is important. The region of flow within

the solution domain with steepest velocity and temperature gradients requires a finer mesh for accurate modeling.

- The type of element used in the scheme affects performance. Considerable accuracy can be achieved with elements and nodes aligned in the flow direction [2, 77].

Real–life problems are usually three–dimensional and thus require three-dimensional simulations. The development of NISUS with 8–node hexahedral elements in previous chapters was motivated by a desire to overcome limitations of previous advection schemes, in terms of stability, accuracy, and applicability.

As discussed previously, iterative solvers are widely used for large–scale problems, mainly because of robustness. Despite the possibility of higher accuracy with direct solvers, they have large memory requirements when large-scale problems are considered [34, 35]. In this chapter, numerical studies of several problems will be considered, in order to assess the performance of NISUS and the new compressed data storage format. These problems include a scalar step–change in convective transport, combined advection and diffusion in an inlet/outlet tank, radial heat flow in a rotating hollow sphere, and others.

6.2 Scalar Step-Change in Convective Transport

A solution domain is discretized with the dimensions of a unit cube. Four cases under consideration are well known, as they were previously studied by Schneider and Raw [4], Hassan et al. [6], and Raithby [7] in 2–D. The boundary values of ϕ (arbitrary scalar) are assigned with reference to the plane *abcd*, i.e., $\phi = 10$ on one side of the

plane, while $\phi = 0$ on the other side of the plane, and $\phi = 5$ at the nodal points where the plane intersects the boundaries (see Fig. 6.1). The transport of ϕ is considered to be pure convection, so the diffusion coefficient is zero (however, all calculations for the problem were performed with $\Gamma = 10^{-10}$). In all of the cases considered, the results were obtained after 15 iterations using a direct solver. The governing equation is

$$\mathbf{v} \cdot \nabla\phi = 0 \tag{6.2.1}$$

with the velocity fields co–planar in the direction of *abcd*.

With the 3–node/ 3–point NISUS, the predicted results ($11 \times 11 \times 11$ mesh) of the scalar, ϕ, along the diagonal *mo* on the plane *mnop* are presented with 4 different values of y_c ($0 \leq y_c \leq 1.0$) (see Figs. 6.1 and 6.2). The upwind differencing scheme exhibits considerable false diffusion across the computational gridline. However, other schemes (not shown in Fig. 6.2a), including Baliga and Patankar [3], and the Positive Coefficient Scheme [4], as well as NISUS, exhibit excellent accuracy. This occurs because of skew upwinding in the schemes and the flow direction is exactly aligned with the plane *abcd*, which is inclined at 45 degrees to the z-axis.

For the case of $y_c = 0.3$ (Fig. 6.2b), there is a noticeable reduction in the level of false diffusion with the upwind differencing scheme. This occurs as a result of the reduction in the level of misalignment, as described above, thereby increasing the weighting on the upstream nodal values according to the gridline. This key advantage of upwind differencing was explored by Prakash [77] using a pseudo-triangulation pattern of domain discretization, which follows the flow direction in order to reduce the error of false diffusion. However, the level of diffusion error is still greater than skew upwinding schemes, i.e., Baliga and Patankar, Positive Coefficient Scheme, and

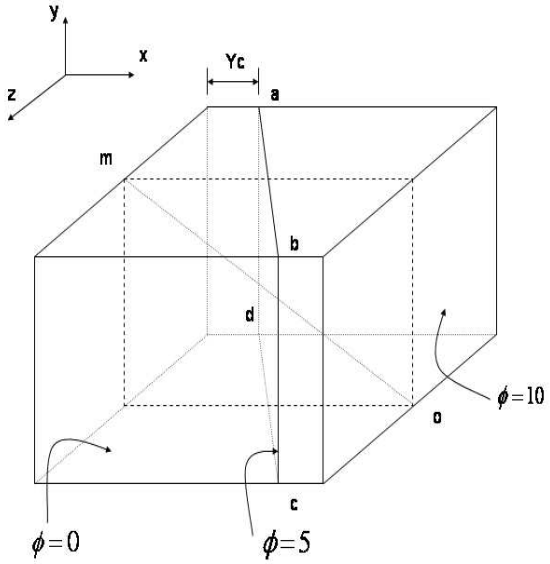


Figure 6.1: Convective transport of a step change along mo with $0 \leq y_c \leq 1.0$

NISUS. A significant difference in the performance of NISUS and the Positive Coefficient Scheme, in terms of false diffusion and avoiding overshoots/undershoots, cannot be observed for this case. However, the scheme of Baliga and Patankar exhibits significant overshoots.

Similar characteristics, as described previously for the case of $y_c = 0.3$, can be observed with $y_c = 0.8$. The level of overshoots of the scheme of Baliga and Patankar become more pronounced. With all cases including $y_c = 1.0$, it appears that the performance of NISUS is comparable with the scheme of Schneider and Raw (i.e., Positive Coefficient Scheme), while showing agreement in terms of accuracy, symmetry with the actual step–change location and numerical stability. Previous schemes (such as the Positive Coefficient Scheme [4]) entail local inversion of the upwind coefficient matrices due to implicitly coupled integration point and nodal variables in

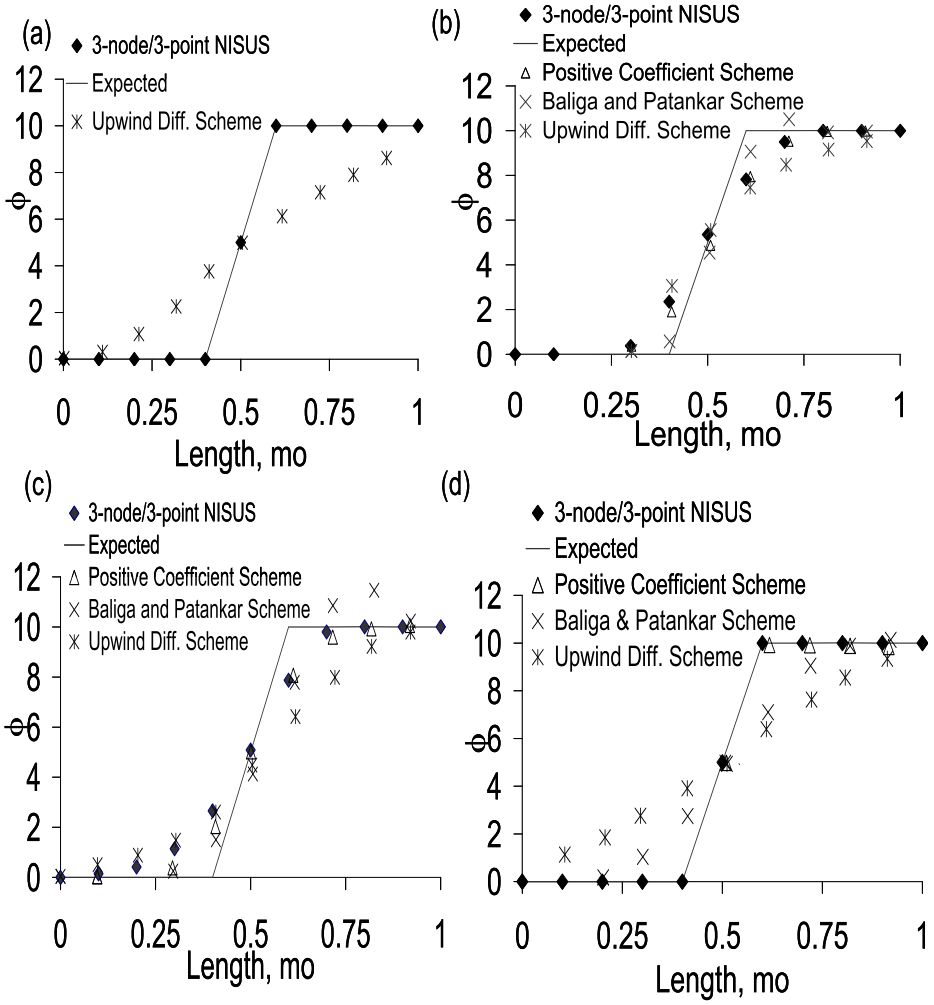


Figure 6.2: Convective transport of a step change in Φ along mo for (a) $y_c = 0.0$; (b) $y_c = 0.3$; (c) $y_c = 0.8$; and (d) $y_c = 1.0$

the upwinding. Since these qualities of NISUS are achieved without local inversion of any convection upwind matrices, it is viewed that NISUS offers a useful advantage in terms of solution economy.

In regards to performance of the two schemes, another scalar step–change problem of convective transport has a solution domain discretized with dimensions of a unit cube. The three cases under consideration have been previously examined by Muir and Baliga [2]. These cases are illustrated in Figs. 6.3 - 6.5, with a uniform velocity field shown as co-planar with *abcd*. The boundary values of ϕ are assigned with reference to the plane *abcd*. In this problem, $\phi = 10$ on one side of the plane, while $\phi = 0$ on the other side of the plane and $\phi = 5$ at the nodal points where the plane intersects the boundaries.

The velocity fields in the three cases are given by

$$\text{case 5:} \qquad \mathbf{v} = \frac{1}{\sqrt{2}}(\hat{i} + \hat{j}) \tag{6.2.2}$$

$$\text{case 6:} \qquad \mathbf{v} = \frac{1}{\sqrt{3}}(\hat{i} + \hat{j} - \hat{k}) \tag{6.2.3}$$

$$\text{case 7:} \qquad \mathbf{v} = \frac{1}{\sqrt{3}}(\hat{i} + \hat{j} + \hat{k}) \tag{6.2.4}$$

In all cases, the converged results were obtained in less than 15 iterations. The predicted results of the scalar, ϕ , along the diagonal *mo* on the plane *mnop* are presented for the three cases in Fig. 6.6. In view of maintaining mesh consistency when comparing results against previous studies, results from an $11 \times 11 \times 11$ mesh are shown in Fig. 6.6. Grids selected for other problems (Figs. 6.9 - 6.16) also reflect this

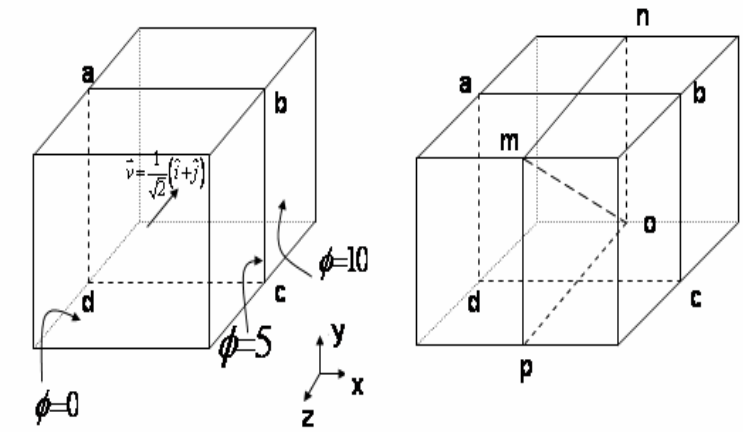


Figure 6.3: Case 5 - Convective step change of temperature along *mo*

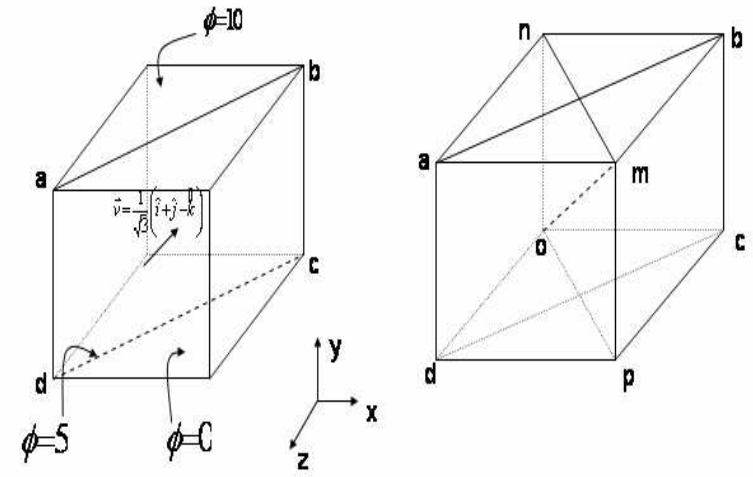


Figure 6.4: Case 6 - Convective step change of temperature along *mo*

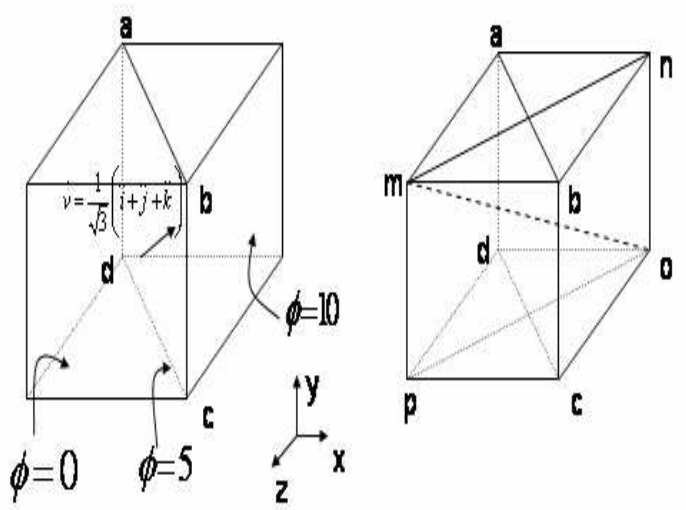
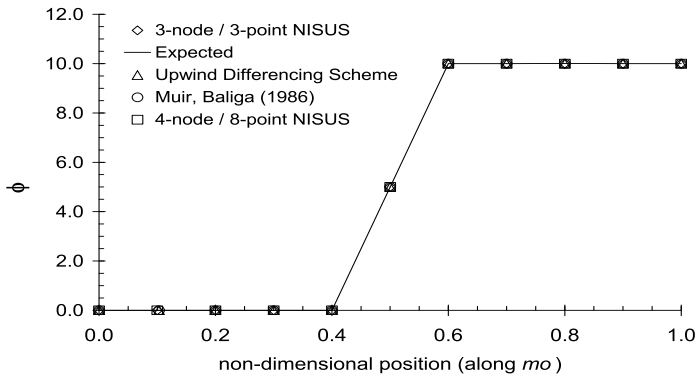


Figure 6.5: Case 7 - Convective step change of temperature along *mo*

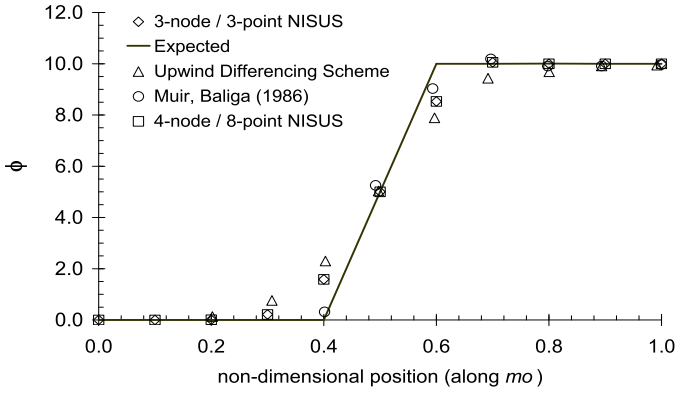
consistency with past studies. For case 5 (Fig. 6.6a), all schemes (UDS, Muir / Baliga, 3–node / 3–point NISUS and 4–node / 8–point NISUS) perform well without false diffusion. This result is expected since the flow streamlines and the computational gridlines are aligned with each other. For cases 6 - 7 in Figs. 6.6b - c, the results of Muir and Baliga [2] agree closely with the proposed scheme in terms of false diffusion, while providing better accuracy than UDS. But their results exhibit overshoots and undershoots for case 6, with slight variations in the levels of these oscillations. The 3–node / 3–point and 4–node / 8–point NISUS schemes exhibit similar performance and accuracy in each case. It can be observed that NISUS exhibits symmetry about the step-change location, without the overshoots and undershoots.

Consider cases 5 and 6 again with the Krylov iterative solvers. In Figs. 6.7 and 6.8, the predicted temperature profiles along the diagonal plane $m - o$ are observed with different iterative solvers and an $11 \times 11 \times 11$ mesh. In the sixth case (Fig.

(a)



(b)



(c)

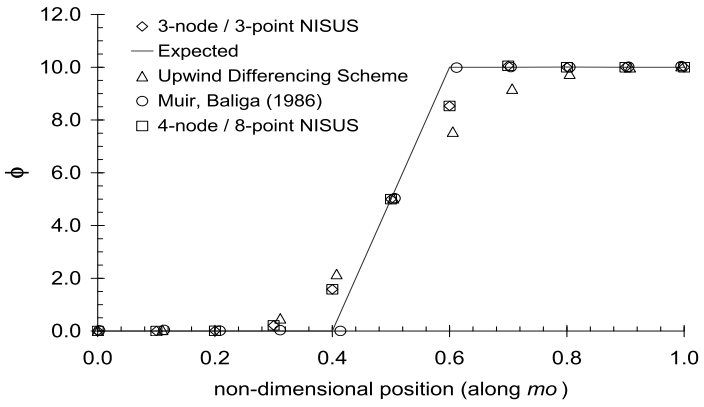


Figure 6.6: Predicted results along *mo* for (a) case 5, (b) case 6 and (c) case 7

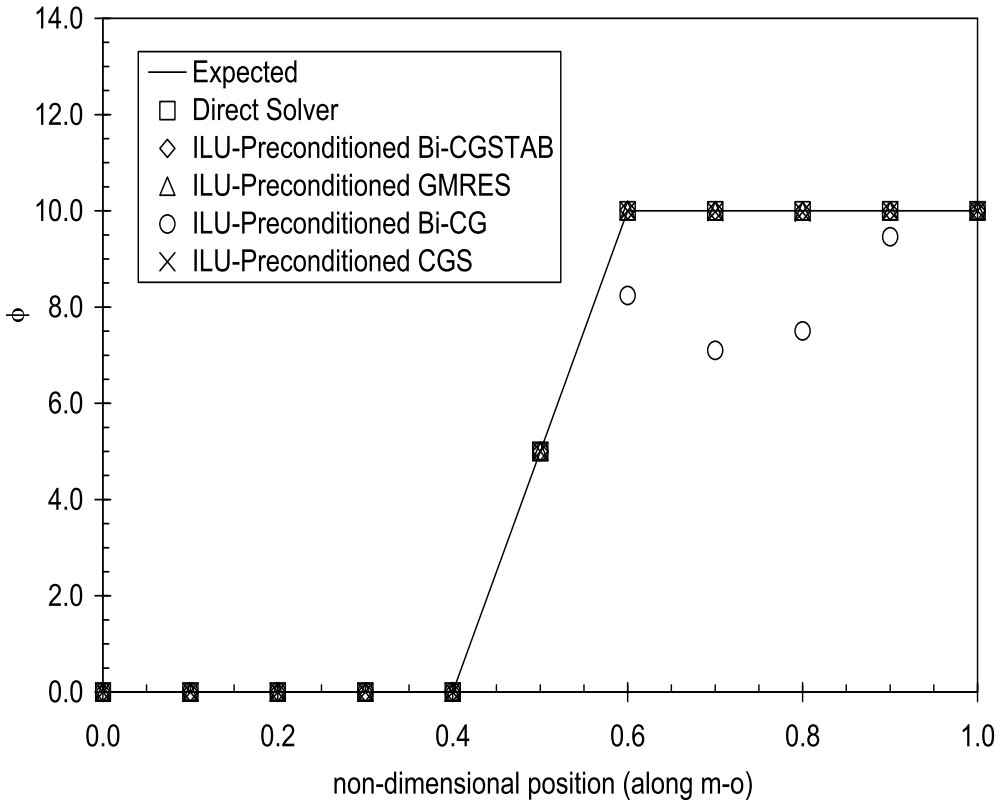


Figure 6.7: Solver performance (case 5: $\mathbf{v} = \frac{1}{\sqrt{2}}(\hat{i} + \hat{j})$)

6.7), where $\mathbf{v} = \frac{1}{\sqrt{2}}(\hat{i} + \hat{j})$ at the inlet, equivalent accuracy is observed for all solvers, except Bi–CG which exhibits an oscillation and poor accuracy on the 10^oC side of the interface. Similarly, Bi–CG yields less accuracy for the second case (inlet velocity of $\mathbf{v} = \frac{1}{\sqrt{3}}(\hat{i} + \hat{j} - \hat{k})$ in Fig. 6.8), in comparison to the other iterative solvers and a direct solver. All schemes have used the same type of convection modeling (NISUS) and numerical discretization (CVFEM), except that different preconditioned solvers

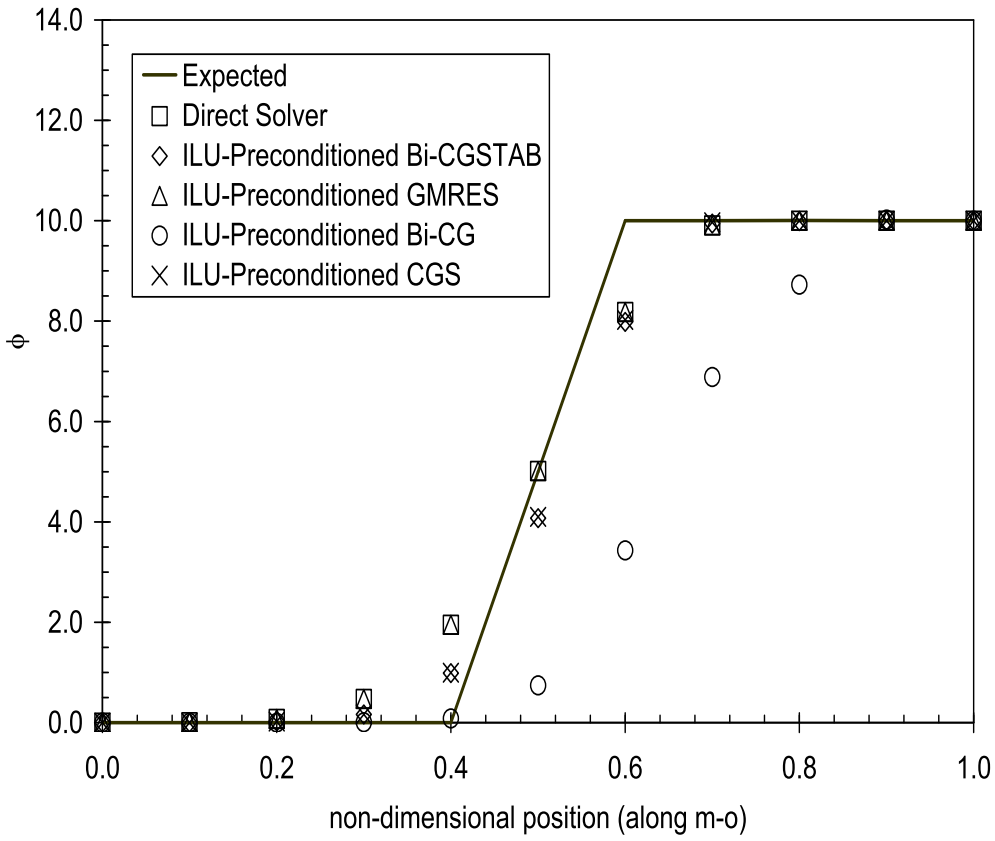


Figure 6.8: Solver performance (case 6: $\mathbf{v} = \frac{1}{\sqrt{3}}(\hat{i} + \hat{j} - \hat{k}\ )$)

were used. These solvers involve approximations to the coefficient matrix with preconditioning, so it appears that Bi–CG yields less accuracy with ILU preconditioning when applied to banded matrices generated from the CVFEM.

6.3 Combined Advection and Diffusion in an Inlet/Outlet Tank

The solution domain for this test problem is discretized into $21 \times 11 \times 11$ nodes. This is a popular benchmark problem examined previously in 2–D by others [4, 5, 78, 79]. The problem geometry and boundary conditions are shown in Fig. 6.9. The governing transport equation becomes

$$\mathbf{v} \cdot \nabla\phi - \frac{1}{Pe}\nabla^2\phi = 0 \tag{6.3.1}$$

where the velocity field and Peclet number are given by

$$\mathbf{v} = 2y(1-(x-1)^2)\hat{i} - 2(x-1)(1-y^2)\hat{j} \tag{6.3.2}$$

$$Pe = \frac{\rho\bar{v}L}{\Gamma} \tag{6.3.3}$$

and the boundary distribution of ϕ at the inlet is given by

$$\phi = 1 + \tanh\left[20(x-1)+10\right] \quad , 0 \leq x \leq 1 \tag{6.3.4}$$

The predicted results of temperature along bc (outlet of the tank) with various Peclet numbers ranging from 10 to 10^6, are shown in Figs. 6.10 and 6.11. The 2–D inlet/outlet problem was investigated by others using various schemes, as reported by Smith and Hutton [5] and Schonaeur [78, 79]. Schneider and Raw [4] showed some comparisons of these various schemes with their positive–coefficient scheme. However, it is noteworthy to observe the steeper and more accurate gradient of the proposed scheme at $Pe = 10^6$, in comparison with other schemes.

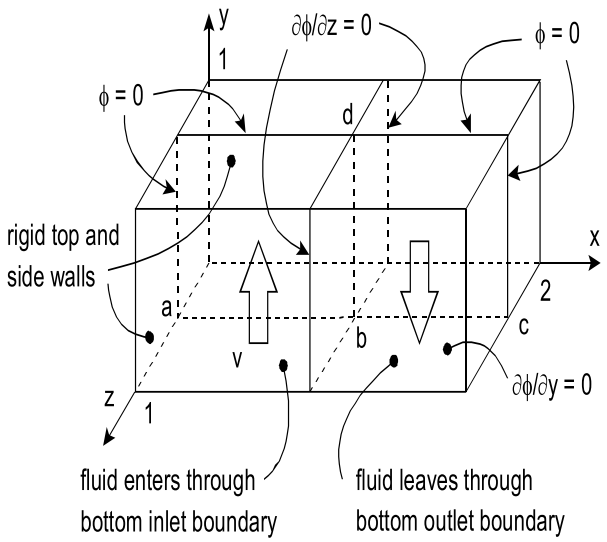


Figure 6.9: Schematic of an inlet/outlet tank

A standard grid for Fig. 6.9 with a uniform nodal spacing of $0.1m$ was adopted for consistent comparisons among various schemes. This configuration was the recommended grid for this problem at the Third Meeting of the International Association for Hydraulic Research Working Group on refined modeling of flows [5]. Similar attempts have been made for obtaining better prediction of the scalar profile at the outlet of the tank at various Peclet numbers, such as Schneider and Raw [4] with the same mesh size. At low Peclet numbers, comparable accuracy is achieved with each scheme. At higher Peclet numbers, the influence of upstream nodal values increases when advection effects become dominant. Performing the convective upwinding without local matrix inversion in the 3–node / 3–point NISUS and the 4–node / 8–point NISUS appears to affect solution accuracy, as well as economy (outlined further in upcoming section).

Accurate prediction of the scalar profile at high Peclet numbers was reported to

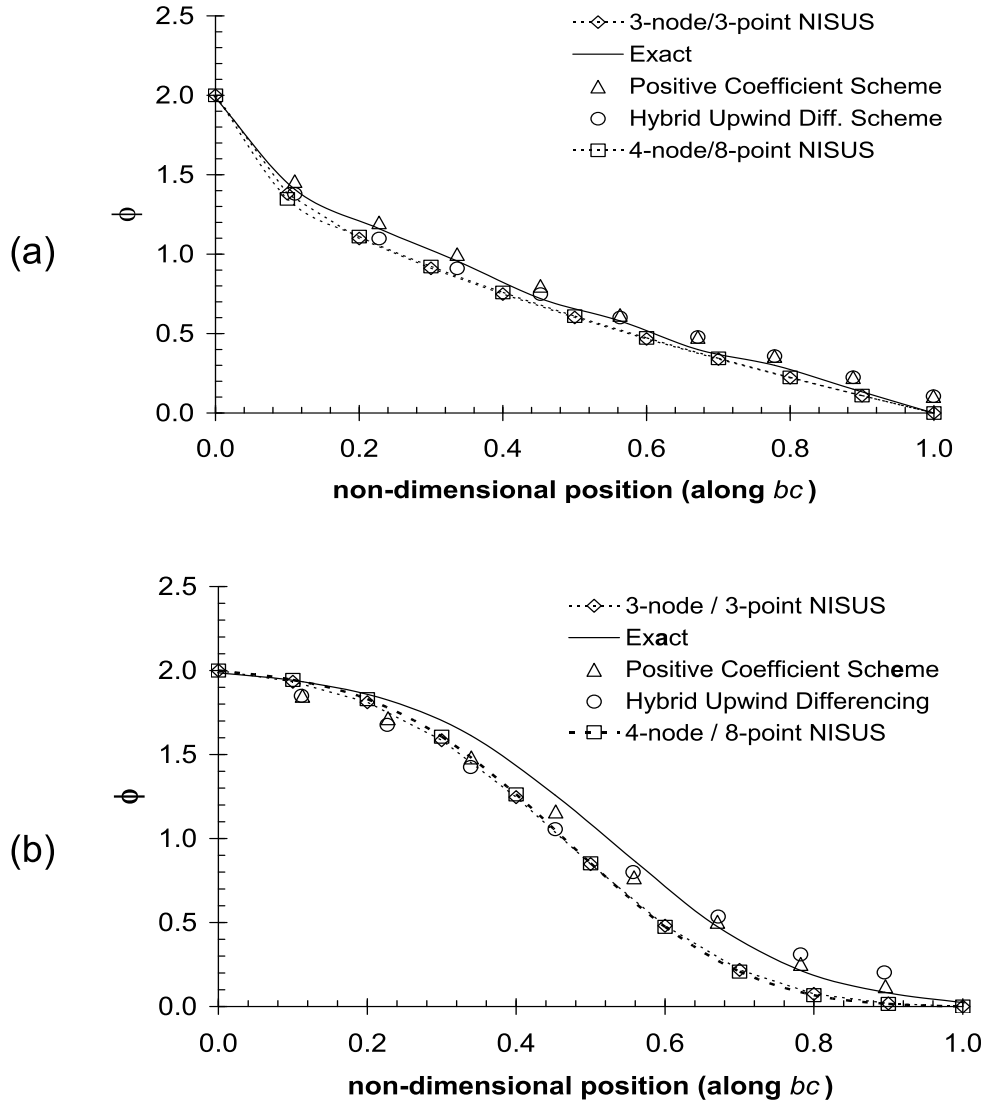


Figure 6.10: Predicted results along bc: (a) $Pe = 10$ and (b), $Pe = 100$

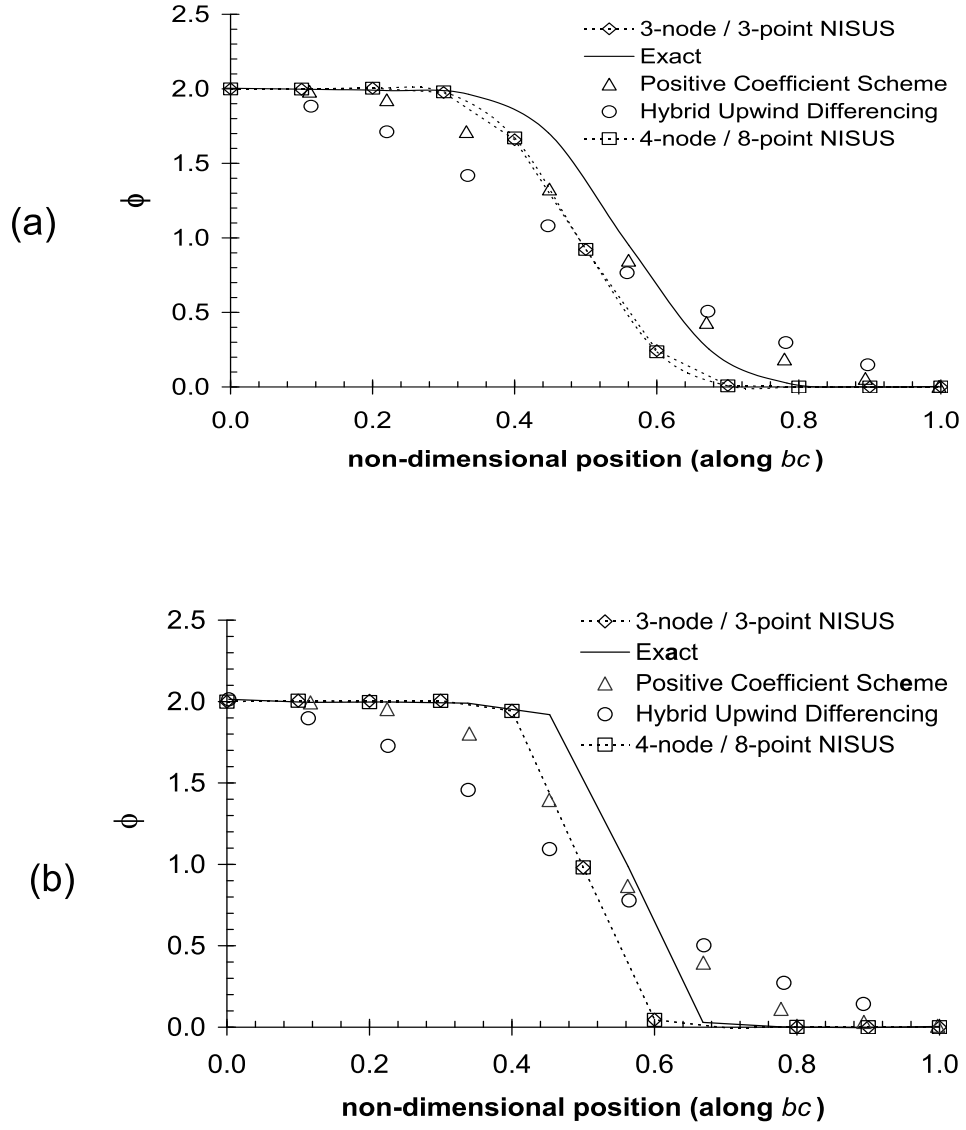


Figure 6.11: Predicted results along bc: (a) $Pe = 500$ and (d) $Pe = 10^6$

be a major challenge by Smith and Hutton [5]. The proposed mass flow weighting of upstream nodes appears to more accurately predict the convected quantity, as compared with combined interpolation of integration point and nodal variables at high Peclet numbers when convection dominates. For example at $Pe = 500$, the steepness of the gradient approaches the exact solution. When matrix inversion in this latter interpolation is performed, additional influences of other nodes through the integration point equation are experienced, which may likely contribute to the level of error experienced at high Peclet numbers, unlike upwind differencing to the nodal points directly in NISUS. Furthermore, the upwind interpolation functions , N_{up}, appear to reduce the upstream differencing error through closer effective proximity between the upwind and integration points.

The hybrid upwind differencing scheme exhibits a higher level of false diffusion when the Peclet numbers increases, while close agreement between both versions of NISUS is observed [80, 33]. But the predicted temperature gradient at the outlet with NISUS exhibits close agreement with the exact solution, while showing better agreement than results predicted by other schemes. Also, the 3–node / 3–point and 4–node / 8–point NISUS are investigated through comparisons between the predicted profile of temperature along *bd*. The predicted results at various Peclet numbers are shown in Fig. 6.12. The profile at the outlet of the tank is repeated along *bd* within the tank at various Peclet numbers. Close agreement between results of both schemes is observed.

Although re–circulation occurs in this problem, the problem presented in the next section involves stronger re–circulation, where differences between different versions of NISUS become more pronounced. Using these direct solver results as a benchmark

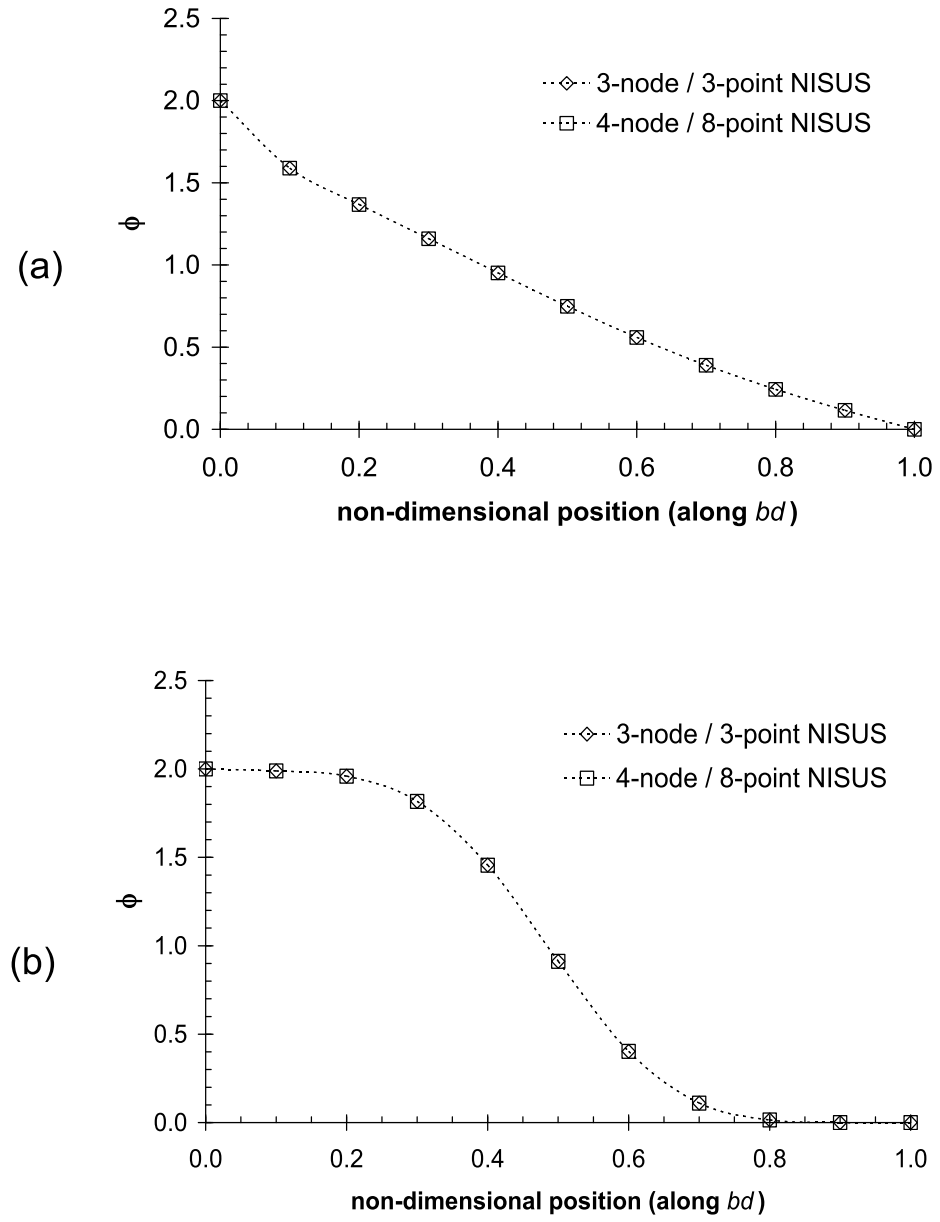


Figure 6.12: Predicted results along bd: (a) $Pe = 10$ and (b), $Pe = 100$

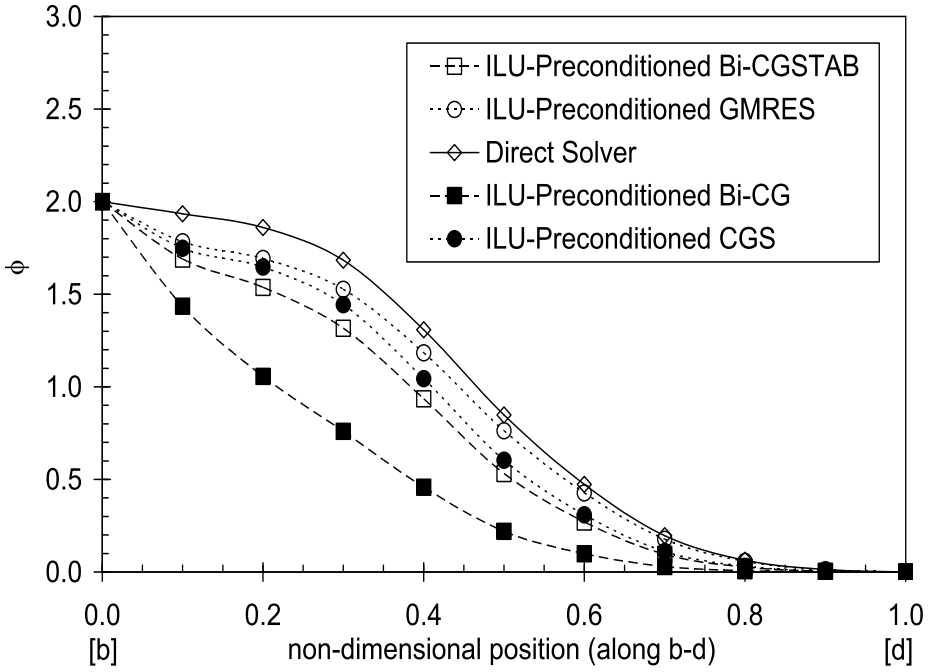


Figure 6.13: Temperature profile at section B-D ($Pe = 100$)

solution, the performance of different ILU–preconditioned Krylov iterative methods (including CGS, Bi–CG, Bi–CGSTAB and GMRES) were investigated. In Fig. 6.13, Bi–CG again shows less accuracy, while the other methods of Bi–CGSTAB, CGS and GMRES exhibit good accuracy. It can be observed that GMRES exhibits better accuracy than Bi–CGSTAB and CGS in Fig. 6.13. The predicted temperature contours are illustrated in Fig. 6.14 for a Peclet number (Pe) of 10. At Pe = 100, the results in Fig. 6.13 show close agreement between the predicted temperature profile (direct solver) and the exact solution.

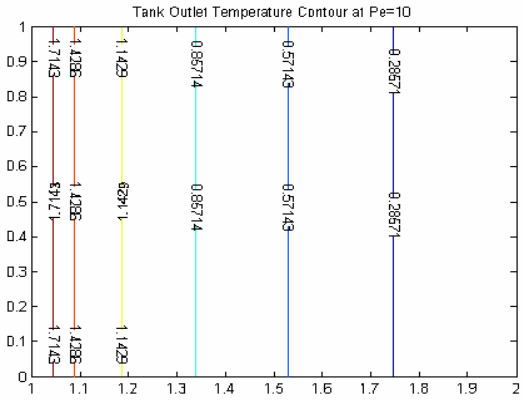


Figure 6.14: Temperature contours at tank outlet ($Pe = 10$)

The performance of the ILU–Preconditioned Krylov iterative methods with respect to data storage, as compared with a direct solver without data compression, is illustrated in Fig. 6.15. Linear curve-fitted approximations of each curve are plotted on the log-log coordinates. The magnification slope of storage savings with the new CBD scheme is up to 51 (see Fig. 6.15). In other words, the CBD format provides over an order of magnitude reduction of storage requirements. This storage savings could allow larger systems (more refined grids) to be solved with a computer having a fixed storage capacity. This advantage of CBD compression becomes particularly significant for three–dimensional heat transfer simulations, when the number of nodal points increases.

6.4 Radial Heat Flow in a Rotating Sphere

This problem is depicted in Fig. 6.16. A hollow sphere with an inner radius, r_1, and outer radius, r_2, in the coordinate system shown, rotates with a constant angular

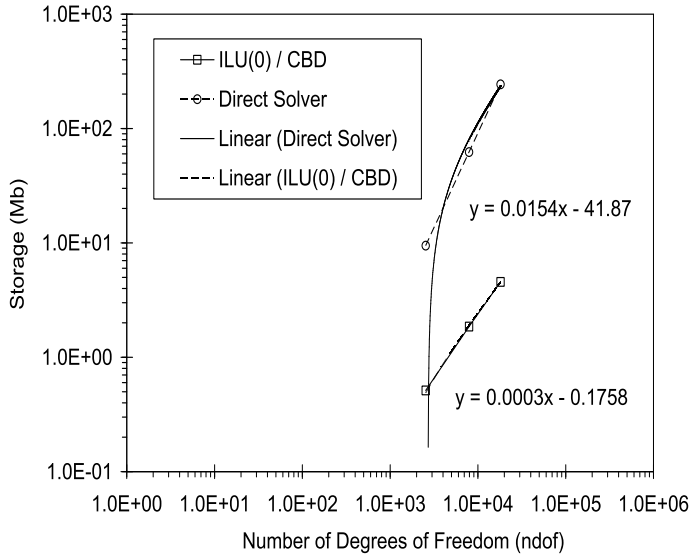


Figure 6.15: Memory requirements for tank flow problem

velocity, $\hat{\omega} = \omega\hat{i}$, about the x–axis. The inner surface of the sphere is maintained at T_1, and the outer surface is kept at T_2. The governing equation is

$$\mathbf{v} \cdot \nabla\phi - \frac{1}{Pe}\nabla^2\phi = 0 \tag{6.4.1}$$

where the velocity field and Peclet number are given by

$$\mathbf{v} = \omega y\hat{k} - \omega z\hat{j} \tag{6.4.2}$$

$$Pe = \frac{\rho(r_1\omega)r_1}{k/c_p} \tag{6.4.3}$$

The boundary conditions are evaluated from the analytical solution as follows [2],

$$\theta = \frac{T - T_2}{T_1 - T_2} = 2\left(\frac{x^2 + y^2 + z^2}{r_1^2}\right)^{-1/2} - 1 \tag{6.4.4}$$

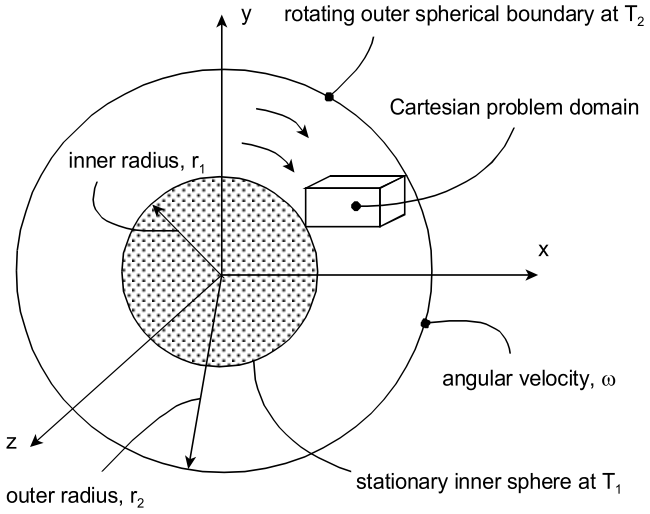


Figure 6.16: Schematic of a rotating hollow sphere problem

A particular case of $r_2/r_1 = 2$ and $T_1 > T_2$ is considered. The results for this problem are represented in terms of the error, e^*, at the centre point of the solution domain as shown in Fig. 6.17. This error, e^*, is given by

$$e^* = \frac{100 \times |\phi_{analytical} - \phi_{numerical}|}{\phi_{analytical}} \tag{6.4.5}$$

The predicted results, using a direct solver, are compared against other schemes (i.e., upwind, hybrid, power–law, and Muir–Baliga) between the range $0.01 \leq Pe \leq 10^4$. The numerical error of the proposed scheme is maintained at an average of about 0.0058 between $0.01 \leq Pe \leq 10^3$, while showing excellent accuracy in comparison with other schemes. All of the other schemes maintained an average error of about 0.0117 up to $Pe = 10$, before experiencing a considerable increase in the error. A lower level of error was demonstrated by the scheme of Muir and Baliga, when compared with the upwind, hybrid, and power-law schemes. The proposed scheme (NISUS) exhibits

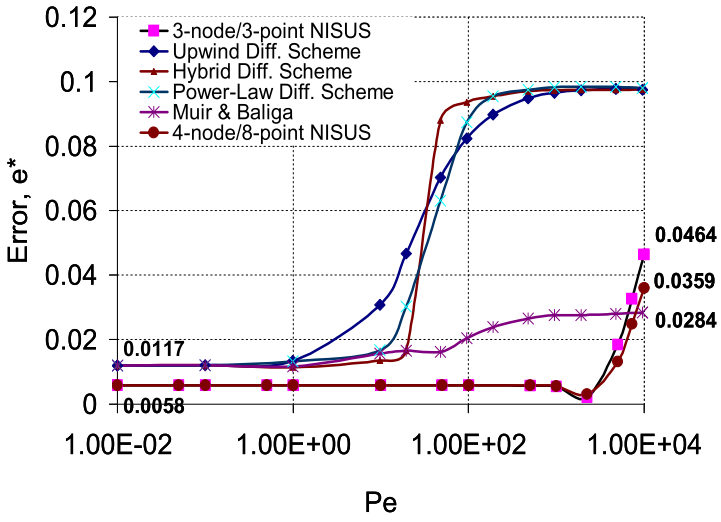


Figure 6.17: Predicted error, e^*, at the center of the solution domain

even better accuracy up to $Pe = 10^4$. Although the error slightly exceeds results of Muir and Baliga, it remains below the other schemes at $Pe = 10^4$. The effect of domain discretization and gridline alignment with the flow direction in obtaining better accuracy has been reported by Prakash [77]. Tetrahedral elements were used in the scheme of Muir and Baliga, thereby exhibiting slightly better accuracy than NISUS at $Pe = 10^4$. NISUS used hexahedral elements, which affects the domain representation in radial flow and becomes more apparent at high Peclet numbers. The performance of the two schemes at $Pe = 10^4$ indicates that the 4–node/8–point NISUS can more accurately predict advection at higher Peclet numbers. Comparisons beyond $Pe = 10^4$ were not presented in the other studies, possibly due to various factors such as turbulence or numerical instability, so Fig. 6.17 is shown only up to that point.

With the iterative solvers, the predicted contours of temperature are illustrated in Fig. 6.18. These results are presented at a Peclet number of $1,000$, while solution accuracy at higher and lower Peclet numbers is shown in Fig. 6.19. Comparing the performance of the ILU-preconditioned iterative solvers with the direct solver (Fig. 6.19), it can be observed that the Conjugate Gradient (CG) Krylov method shows good accuracy up to $Pe = 1$, but it diverges thereafter. Other variants of the CG base (including Bi–CG, CGS, and Bi–CGSTAB) also showed similar characteristics. An equivalent residual tolerance (10^{-7}) was adopted in each case, so it appears that this tolerance becomes insufficient for the preconditioned matrices generated by CG at high Peclet numbers, when the impact of non–linear convection terms increases relative to diffusion. In contrast, the ILU–preconditioned GMRES(10) Krylov method shows consistently good accuracy, with an average of 7 iterations at a residual tolerance of 10^{-7}. Less error is obtained with ILU–GMRES up to Pe = 1, but the error increases between $Pe = 1$ and $Pe = 10$. Nevertheless, the magnitude of this error is very small (below 0.2%).

An advantage of using CBD format compression with the iterative solver is shown in Fig. 6.20. The storage requirements of CBD / ILU(0) are compared against the direct solver in Fig. 6.20, where the magnification slope of storage savings is up to 53 in this problem. The number of degrees of freedom refers to the number of nodal points in the grid. In addition to reduced storage requirements, dense matrix storage can greatly improve CPU run–time in numerical heat transfer simulations. For example, when matrices fit within the storage limits of cache memory of the computer, the simulations can achieve close to the advertised peak CPU speed of the computer processor. However, if the matrices cannot fit within this cache memory limit for large 3-D heat transfer simulations (without data compression), the simulations are

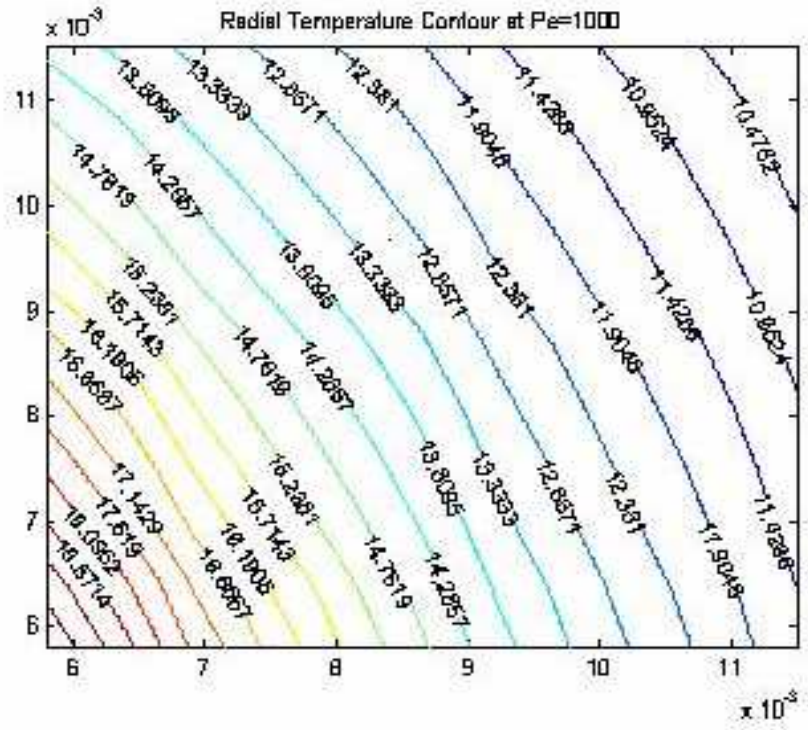


Figure 6.18: Temperature contours ($Pe = 1,000$)

reduced to about 10% of the peak CPU speed [81], depending on the limitations of bus speed within the computer.

Consider an example when the matrices can be stored within cache memory, thereby allowing dense matrix / vector multiplications to be performed in about 3.5 GFlops [81]. In this case, the CPU efficiency falls to about 700 Mflops when matrices become too large to fit within cache memory. This reduced rate of 700 MFlops corresponds to a rate near the bus speed, while the accelerated rate of 3.5 MFlops is higher than half of the advertised CPU peak rate of 6.1 GFlops. If zeroes

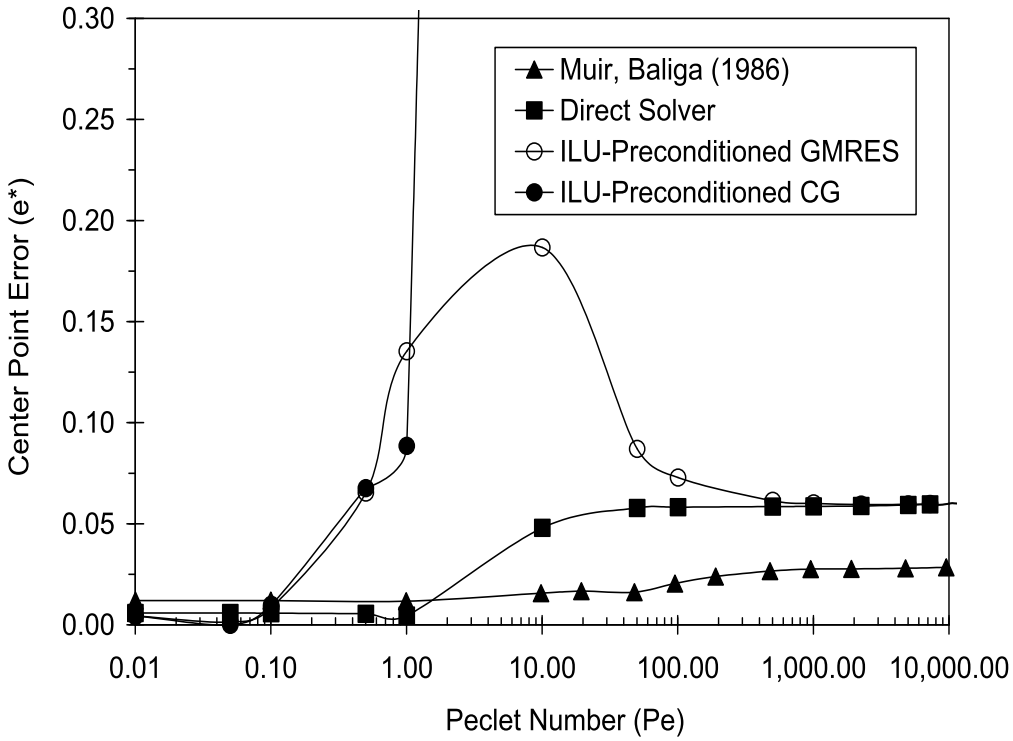


Figure 6.19: Solver performance for radial flow problem

are retained within the matrix structure (without data compression), MFlop rates for sparse matrix / vector multiplications are significantly lower. For example, the rate is significantly reduced to 158.6 MFlops for a matrix size of 100, with 10 non–zero entries that are randomly distributed per column (simulations performed on a Pentium IV / 3.06 GHz Xeon laptop computer [81]). The MFlop rate stays relatively constant at this rate, until the matrix sizes reach about $10,000$ (120 MFlops). At these larger matrix sizes, reading values into memory sequentially flushes the cache memory, so numerous entries must be re–read from cache, which must be refreshed before the beginning of each iteration in the simulations. If the matrix size reaches

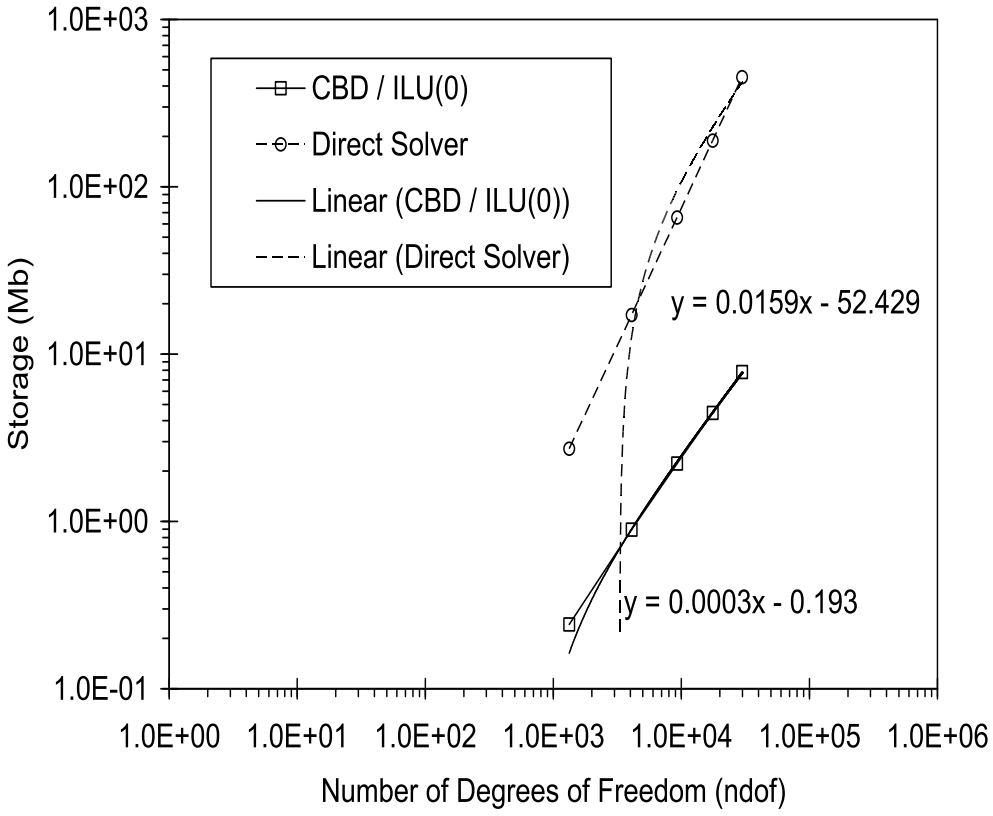


Figure 6.20: Storage requirements for radial flow problem

$1,000,000$, most matrix entries are retrieved from RAM, so the computing speed is reduced drastically to about 18 MFlops.

Thus, data compression is an important consideration from perspectives of both storage requirements and CPU run-time, particularly with three–dimensional heat transfer simulations. The majority of data generated in such simulations is often stored in matrices. As a result, an efficient simulation tool must utilize some type

of data compression, together with a compression format suitable for matrices generated by the particular type of numerical scheme. The previous results have shown the importance of effective data compression for three specific examples, while other applications would also benefit from optimal storage of coefficient matrices.

The range of errors depicted in Fig. 6.17 shows generally good accuracy of NISUS, while exhibiting the added advantage of computational savings. Steady state calculations with a time-step of $10,000$ sec were used with 15 iterations for this problem. A Pentium IV, 2.53 GHz (512 MB RAM) system was used for the simulations. The CPU run-time with different grid sizes is illustrated in Fig. 6.21, which compares NISUS (without local inversion) to the same skew upwind scheme, but requiring an added inversion of a local 12×12 matrix for each element. This inversion refers to the upwind coefficient matrix inverted to represent the integration point variables solely in terms of nodal variables. The results show that there is a time-gain factor (i.e., ratio of the CPU time with local inversion, to the CPU time without local inversion) of about 2.173 for an $11 \times 11 \times 11$ node mesh. This time–gain factor increases quickly as the number of nodal points increases. The performance of the 3-node / 3-point and 4-node / 8-point NISUS, in terms of CPU run-time with different grid sizes has been investigated (see Fig. 6.22). The two schemes (without local inversion) were compared with the same skew upwind procedure, but requiring local inversion of the upwind coefficient matrices. It appears that considerable CPU time savings can be realized with NISUS, without an appreciable loss of accuracy. Since the number of grid points increases quickly in three–dimensional simulations, this feature becomes particularly significant in 3–D problems.

A useful effort in this research is reducing the computational time of convective

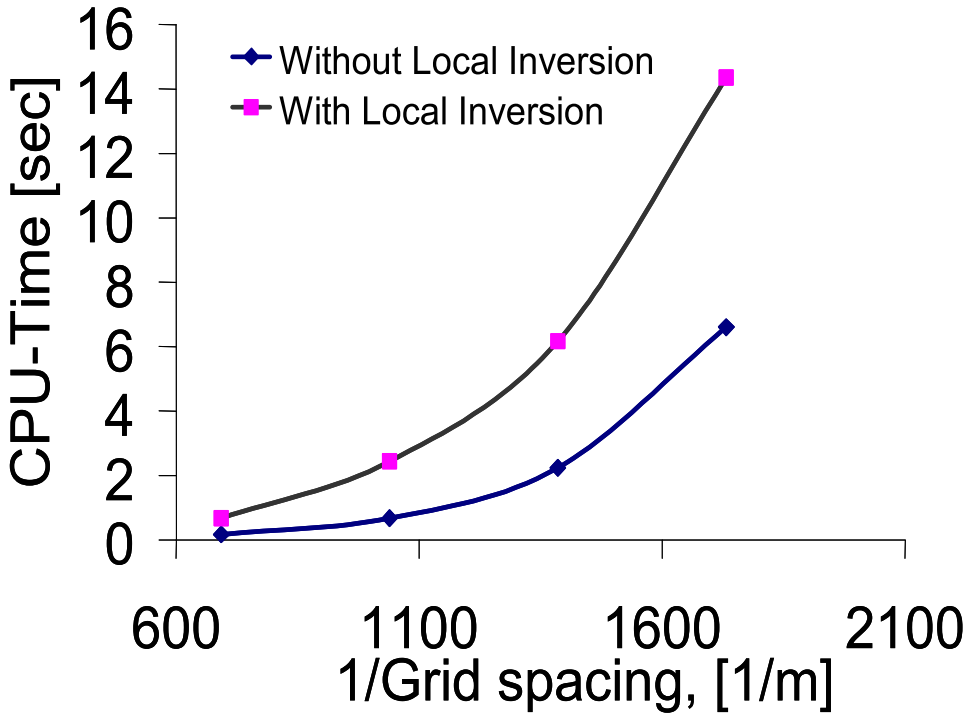


Figure 6.21: CPU time at various grid spacings

upwinding by avoiding the coefficient matrix inversion. It is sufficiently general, so that it can be applied within other types of collocated finite volume schemes. An upwind scheme requires interpolation along the upstream edge for the upwinded value, which usually involves two variables at the adjacent nodes (or four nodes in 3–D problems). NISUS also performs this upwind interpolation, but using mass weighting of nodal values and an integration point equation. Both schemes explicitly represent the upwinded value in terms of nodal values, so their computational expense is considered to be close and small, relative to larger expenses associated with a matrix inversion. Various benefits are realized by the integration point equation

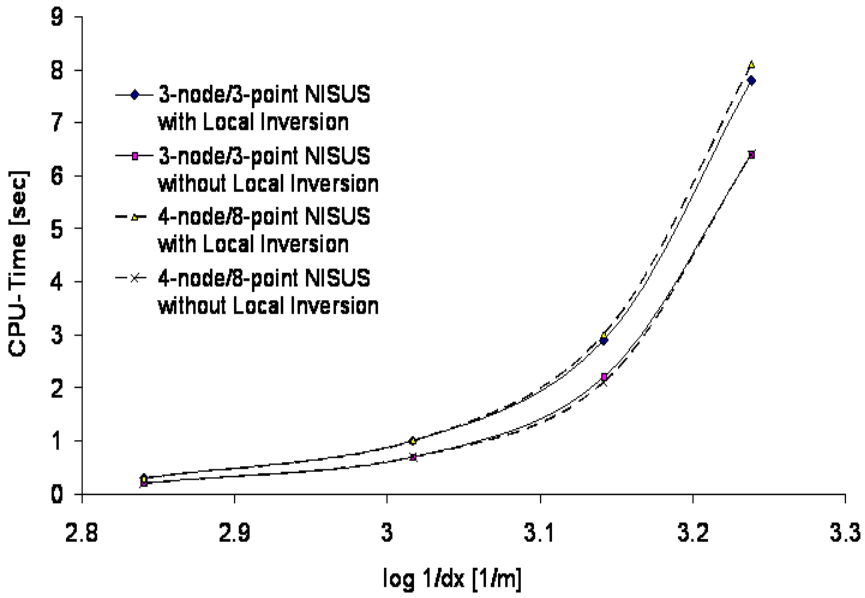


Figure 6.22: CPU performance of NISUS

(as documented by Schneider and Raw [4]), i.e., collocated scheme without requiring staggered grids, while retaining a physically based coupling between problem variables at a sub–element level. Thus, comparable expense and the benefits of integration point modeling are considered to give NISUS a useful alternative to past methods.

Accurately predicting sharp temperature and velocity gradients can be accomplished by a clustered or adaptively refined grid near the location of the sharp gradient. For example, better accuracy in the previous problem may have been obtained by clustering of nodes near the sharp gradient. In general, better accuracy can be obtained by such methods, but at certain expense. CPU time increases and the location of the sharp gradient is often unknown. In that case, adaptive grid refinement near

the sharp gradient may require re-meshing and iterations within a time step, thus considerably increasing the computational expense. Furthermore, the different methods are intended to be compared against each other in terms of convective upwinding accuracy alone. Grid refinement and nodal clustering reflect the performance of the entire algorithm when trying to improve solution accuracy, rather than convection modeling alone.

6.5 Natural Convection in Tilted Enclosures

In the previous sections, results from NISUS and CBD storage were presented for the finite element formulation (CVFEM). In contrast, this section and section 6.6 extend NISUS to the finite volume method with a staggered grid. The purpose is demonstrating that NISUS has valuable utility in both finite element and finite volume methods. Two-dimensional natural convection of a Boussinesq fluid in a tilted cavity of length, L, and height, H, with differentially heated side walls is studied in this section. Rayleigh numbers of 10^3,10^4,10^5 and 10^6 and tilt angles of 0^o (vertical) to 20^o are studied. Using the NISUS-variant EDS as discussed in section 4.3, the computed flow distributions, temperature profiles and heat transfer predictions will be shown to compare favourably with the conventional EDS and benchmark solution.

The velocity boundary conditions are (see Fig. 6.23)

$$\begin{aligned} u(\pm\frac{L}{2}, y) &= v(\pm\frac{L}{2}, y) = 0 \\ u(x, 0) &= v(x, 0) = 0 \\ u(x, H) &= v(x, H) = 0 \end{aligned}$$

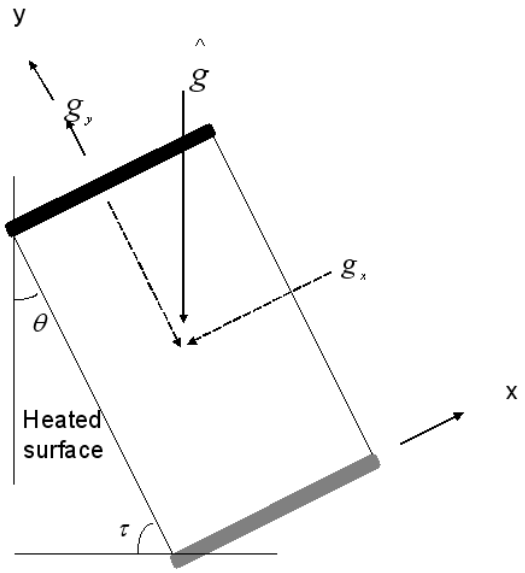


Figure 6.23: Cavity geometry and coordinate system

The temperature boundary conditions are

$$\begin{aligned} T(-\frac{L}{2}, y) &= 12 \\ T(\frac{L}{2}, y) &= 2 \\ \frac{\partial T}{\partial y}|_{x,0} &= 0 \\ \frac{\partial T}{\partial y}|_{x,H} &= 0 \end{aligned}$$

When evaluating results in these problems, a convergence criterion of $|(\phi^{k+1} - \phi^k)/\phi^{k+1}| \leq \epsilon_s$, will be used for the velocity fields, where $\epsilon_s = \epsilon_{u,v} = 10^{-4}$. When the residuals decline below the indicated tolerance, solution convergence is established.

An enclosed cavity (Fig. 6.23) with fluid properties shown in Table 6.1 is considered. The results predicted with the NISUS variant of EDS are compared with the benchmark solution and the conventional EDS are shown in Tables 6.2-6.5.

Table 6.1: Square Cavity Size and Fluid Properties

Fluid	ρ	μ	β	Pr
properties	1.25388012	1.7469×10^{-5}	0.003569134128	0.771502536
Cavity	$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$	$Ra = 10^6$
size (L/H)	0.009201	0.019823	0.042710	0.092015

Table 6.2: Solutions at $Ra = 10^3$

Scheme	u_{max} (y)	v_{max} (x)	Nu_o	Nu_1	$\bar{Nu}$
EDS	3.673695 (0.8250)	3.764558 (0.1750)	1.061103	1.060693	1.119295
NISUS	3.624875 (0.8250)	3.712130 (0.1750)	1.058803	1.058446	1.116796

Table 6.3: Solutions at $Ra = 10^4$

Scheme	u_{max} y	v_{max} x	Nu_o	Nu_1	$\bar{Nu}$
.EDS	16.110555 0.8250	20.255491 0.1250	2.042514	2.036434	2.243852
NISUS	15.621260 0.8250	19.668755 0.1250	2.023959	2.018312	2.223377

Table 6.4: Solutions at $Ra = 10^5$

Scheme	u_{max} y	v_{max} x	Nu_o	Nu_1	$\bar{Nu}$
EDS	33.963022 0.8750	73.381774 0.0750	3.701167	3.701148	4.422836
NISUS	33.406764 0.8750	72.097121 0.0750	3.685402	3.688937	4.40760

Table 6.5: Solutions at $Ra = 10^6$

Scheme	u_{max} y	v_{max} x	Nu_o	Nu_1	$\bar{Nu}$
EDS	83.230471 0.9250	324.086636 0.0250	5.805055	5.655509	7.559999
NISUS	81.815378 0.9250	318.897264 0.0250	5.796038	5.651280	7.556801

Table 6.6: Prediction of Nusselt number as a function of the Rayleigh number

Method	$Ra = 10^3$	$Ra = 10^4$	$Ra = 10^5$	$Ra = 10^6$
de Vahl Davis	1.118	2.243	4.519	8.800
EDS	1.119295	2.243852	4.422836	7.559999
(Error%)	(0.116)	(0.038)	(2.128)	(14.091)
NISUS	1.116796	2.223377	4.40760	7.556801
(Error%)	(0.108)	(0.875)	(2.465)	(14.127)

The predicted Nusselt number, with the NISUS variant of EDS favourably agrees with de Vahl Davis's [82] bench-mark solution, as shown in Table 6.6. Both EDS and NISUS have comparable accuracy over a range of Rayleigh numbers, although larger errors are observed at high Rayleigh numbers. By comparison with the conventional EDS, better accuracy is observed at $Ra = 10^3$, while the NISUS variant of EDS maintains close accuracy at higher Rayleigh numbers. It is observed that the performance of the conventional EDS over the NISUS variant of EDS increases with increasing Rayleigh number. NISUS can better predict flows at lower Peclet (and by extension, Rayleigh) numbers [15], especially with radial flow streamlines. However, both the conventional EDS and NISUS variant of EDS exhibit significant increases in error at $Ra = 10^6$, apparently because of turbulence effects.

Numerical simulations were conducted with various grid refinements and only grid-independent results will be shown, as negligible changes in solution accuracy occurred with finer grids. Predicted temperature contours at $Ra = 10^4$ are illustrated in Figs. 6.24 - 6.25 for a 20×20 mesh. Similar results are obtained for both NISUS (Fig. 6.24) and EDS (Fig. 6.25). Tilt angles of 5^o and 20^o are considered for both cases. It can be observed that the isotherms are slightly more distorted at the lower tilt angle. The vertical component of gravity is larger at the lower tilt angle, so higher buoyancy

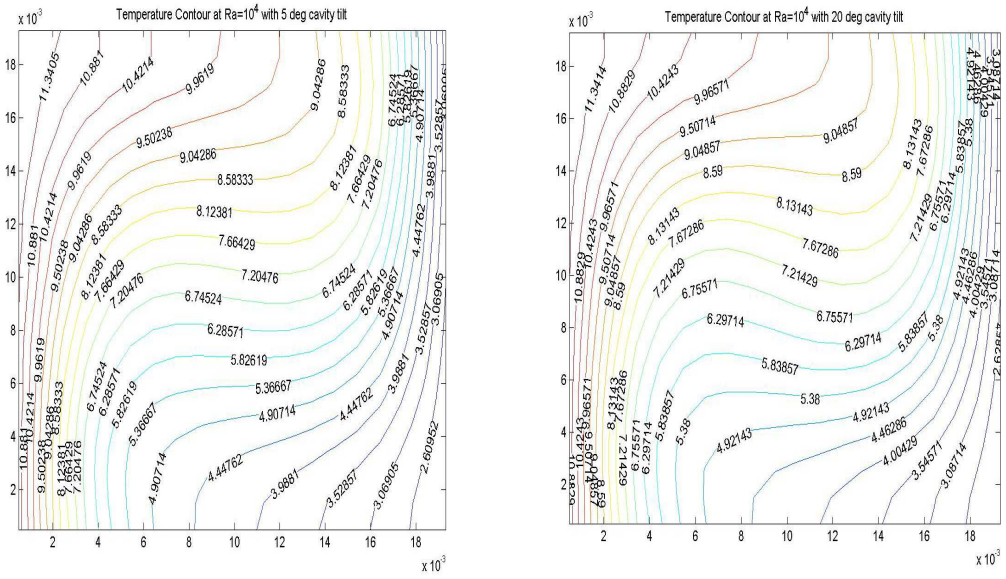


Figure 6.24: Temperature contours at different tilt angle with NISUS

forces and fluid velocities distort the temperature fields differently. The temperature contours (Figs. 6.25-6.24) at $Ra = 10^4$ reveal a slight skewness in the hot side to the corner of the cavity, as the tilt angle increases.

In Fig. 6.26, the performance of NISUS and EDS is compared with respect to solution residuals. The values R(U) and R(V) refer to the U-velocity and V-velocity residuals, respectively. The residual decline can be observed as more monotonically uniform for NISUS, as compared with EDS in Fig. 6.26(a). In Fig. 6.26(b), both residuals decline monotonically, although NISUS reaches its asymptote sooner. In the solution procedure, each momentum equation is solved separately. Inter-equation iterations are performed until the variable change from one iteration to the next iteration becomes sufficiently small, so the solution is considered converged.

At convergence, all discrete equations are satisfied in all control volumes to the

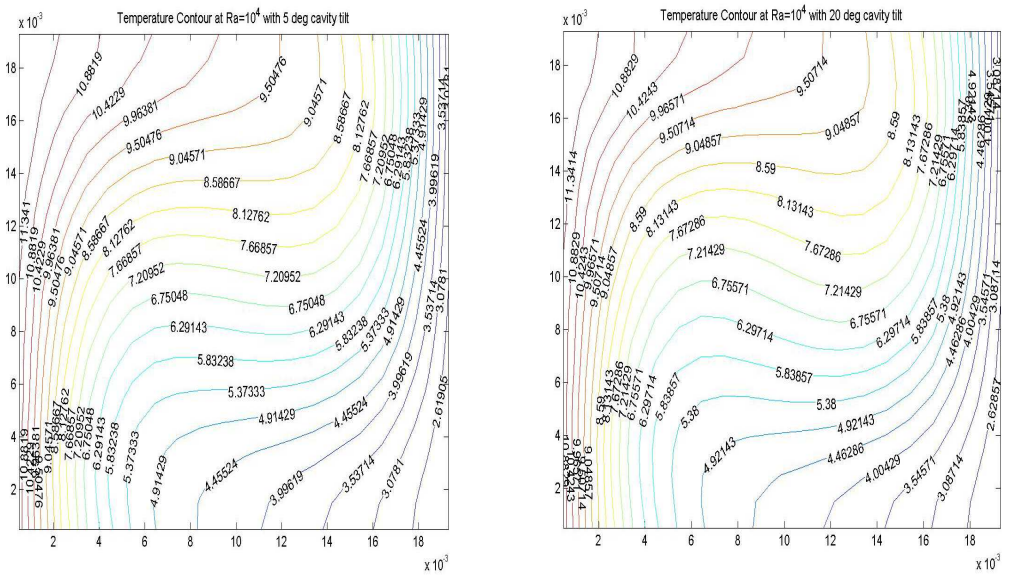


Figure 6.25: Temperature contours at different tilt angle with EDS

specified tolerance level and the solution no longer changes with additional iterations. During the inter-equation iterations, the residuals measure the imbalance (or error) in the discretized governing equations. Solutions are generally converged when the flow field is no longer changing. If the residuals have satisfied the specified convergence criterion, but they are still decreasing, then the solution may not yet be converged. Conversely, if the residual is no longer decreasing, but it does not satisfy the convergence criterion, then it may have converged. Also, low residuals do not automatically imply a correct solution, while high residuals do not automatically imply an incorrect result. In the current results, small residuals were not changing after about 150 iterations. Converged solutions were obtained at approximately that point. With residuals declining more monotonically for NISUS in Fig. 6.26(a), the upstream weighting on flow directionality has beneficial effects on better stability of

convective non-linearities.

6.6 Microchannel Flows

Microchannel flows can be divided into different flow regimes that require different methods of analysis, depending on the Knudsen number (Kn). Figure 6.27 illustrates a gaseous microchannel flow with a slip boundary condition. The ratio of the mean free path of the fluid to a characteristic length scale of the problem is called the Knudsen number. The Navier-Stokes equations with a slip-flow boundary condition can be used in the regime of $0.001 < Kn < 0.1$. Incompressible flow of gas will be studied and the ideal gas law ($P = \rho RT$) will be used, with constant properties of ρ, R and μ (density, gas constant and dynamic viscosity, respectively) throughout the channel. Both no-slip and slip boundary conditions in the microchannel will be considered. In the latter case, the following first-order wall-slip velocity model and associated parameters from Ref. [83] will be used,

$$u_{gas} - u_{wall} = \xi_1 \frac{\partial u}{\partial y}|_{wall} + \xi_2 \frac{\partial T}{\partial x}|_{wall} \tag{6.6.1}$$

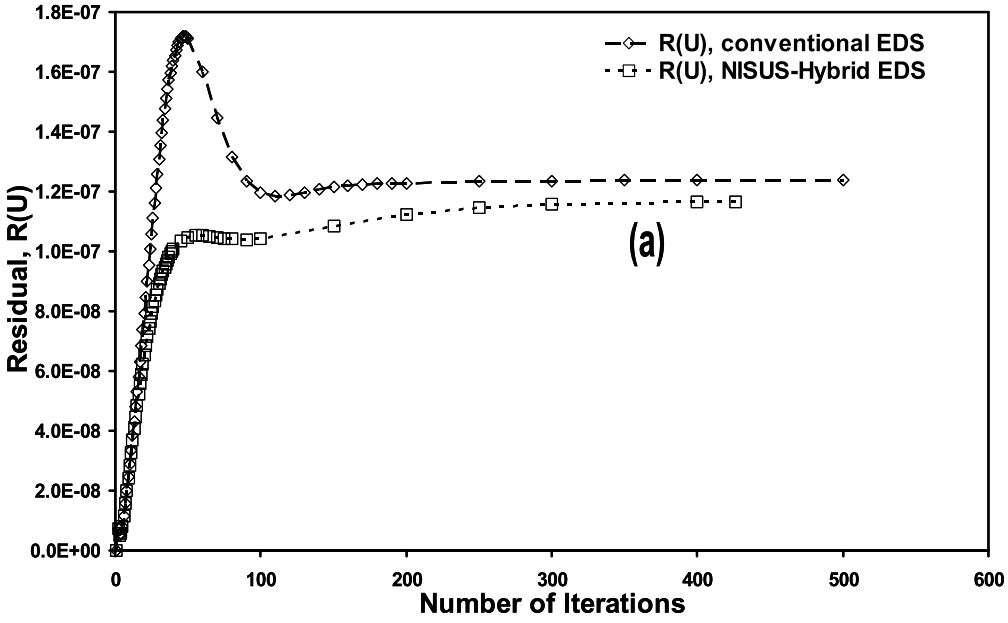
where

$$\xi_1 = \frac{2 - \sigma}{\sigma} \lambda \tag{6.6.2}$$

and

$$\xi_2 = \frac{3\mu}{4\rho T_{gas}} \tag{6.6.3}$$

In this equation, σ and λ refer to the tangential momentum accommodation coefficient (or TMAC) and mean free path, respectively. The TMAC value characterizes the fraction of molecules reflected diffusively from a wall. At high Knudsen numbers,



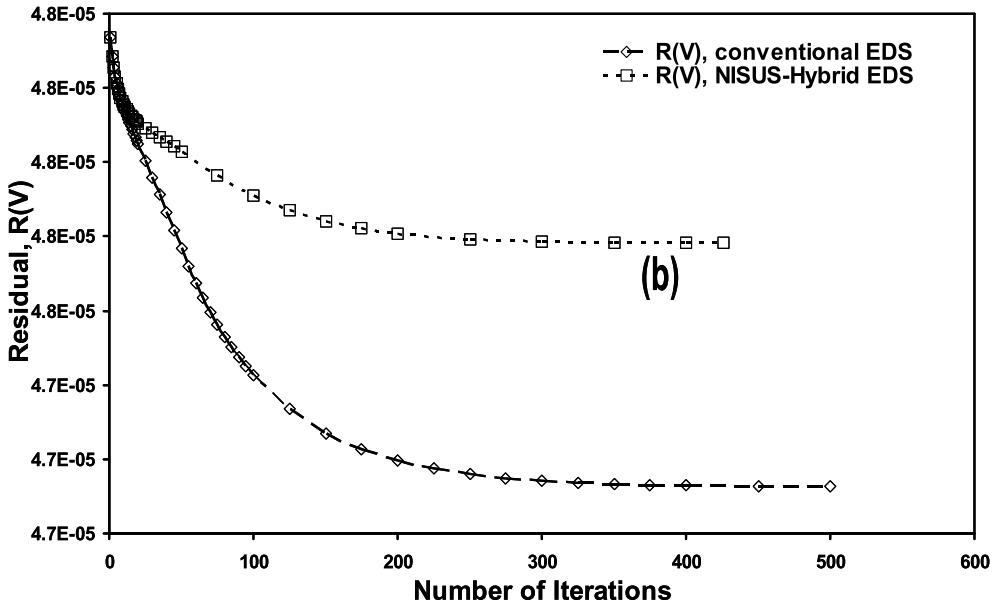


Figure 6.26: Residual analysis of the natural convection solution field; (a) U-field; (b) V-field

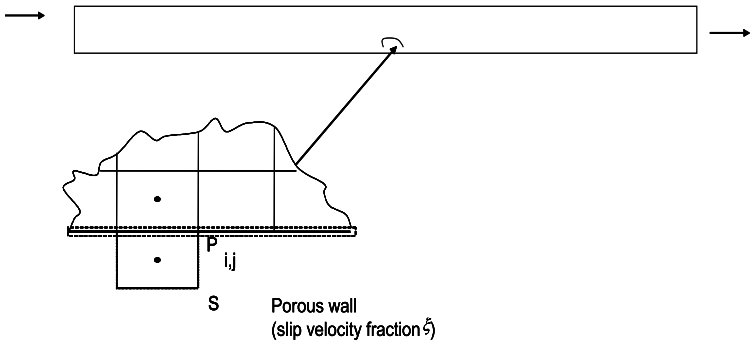


Figure 6.27: Porous boundary of a channel medium

intermolecular interactions with the wall represent an increasing portion of the total intermolecular exchanges. A differential control volume in the continuum equations approaches the thickness of a near-wall layer containing this intermolecular exchange, so a non-zero velocity is obtained by momentum-averaging of the near-wall collisions. Nitrogen gas is used as the working fluid (see Table 6.7) and flow simulations are conducted at different pressure ratios. The performance of a new NISUS-Hybrid EDS scheme is validated with analytical and previous numerical results.

Problem parameters for gas flow in a microchannel are shown in Table 6.7. Predicted results for both no-slip and slip flow cases are illustrated in Fig. 6.28(a). It can be observed that the predicted no-slip results compare well with the analytical solution. This provides useful validation for the fluid flow and NISUS formulations. These results are consistent with the solutions of the linearized Boltzmmann equations and the DSMC, as reported by Xue and Fan [84]. Comparing the results of the no-slip flow with a slip boundary condition, at the boundary pressure ratio (i.e., p_{out}/p_{in})

Table 6.7: Microchannel dimensions and fluid properties

Flow parameters	Description
Length, L (μm)	$1560, 2560, 3560, 4560, 5560$
Height, H (μm)	1.0
Dynamic viscosity, (Ns/m^2)	0.000016
Specific gas constant, c_p (J/KgK)	296.8
TMAC for velocity, σ_V	0.8
TMAC for temperature, σ_T	0.8
Outlet Pressure, P_{out} (Pa)	100800.0
Pressure ratio, P_{in}/P_{out}	$1.34, 2.70, 3.00, 3.34$
Knudsen number, Kn	0.0579

of 1.34, the maximum streamwise velocity of the no-slip flow is higher than that of slip flow by about 7% (see Figure 6.28(a)). In Fig. 6.28(b), the number of iterations to convergence and resulting CPU run-time are higher for the no-slip case. Greater effort is required to overcome higher gradient of velocity and temperature due to no-slip friction effect and this has direct influence on the number of iterations required before convergence. Consequently, Fig. 6.28(b) presents higher residuals for no-slip conditions, as a result of higher near-wall pressure gradients in this case. Although, the performance of the two upwinding schemes in term of accuracy is the same for the microchannel simulations, it is observed that NISUS converges faster than the conventional EDS (see Fig. 6.28(b)). More fluid kinetic energy is required to overcome friction with a no-slip boundary condition, when compared against slip-flow conditions in microchannel flows.

The performance of EDS and NISUS are nearly identical, with respect to the accuracy of the fully developed velocity profile. A higher error is observed close to the wall of the microchannel as shown in Fig. 6.29. This error, on the average, is

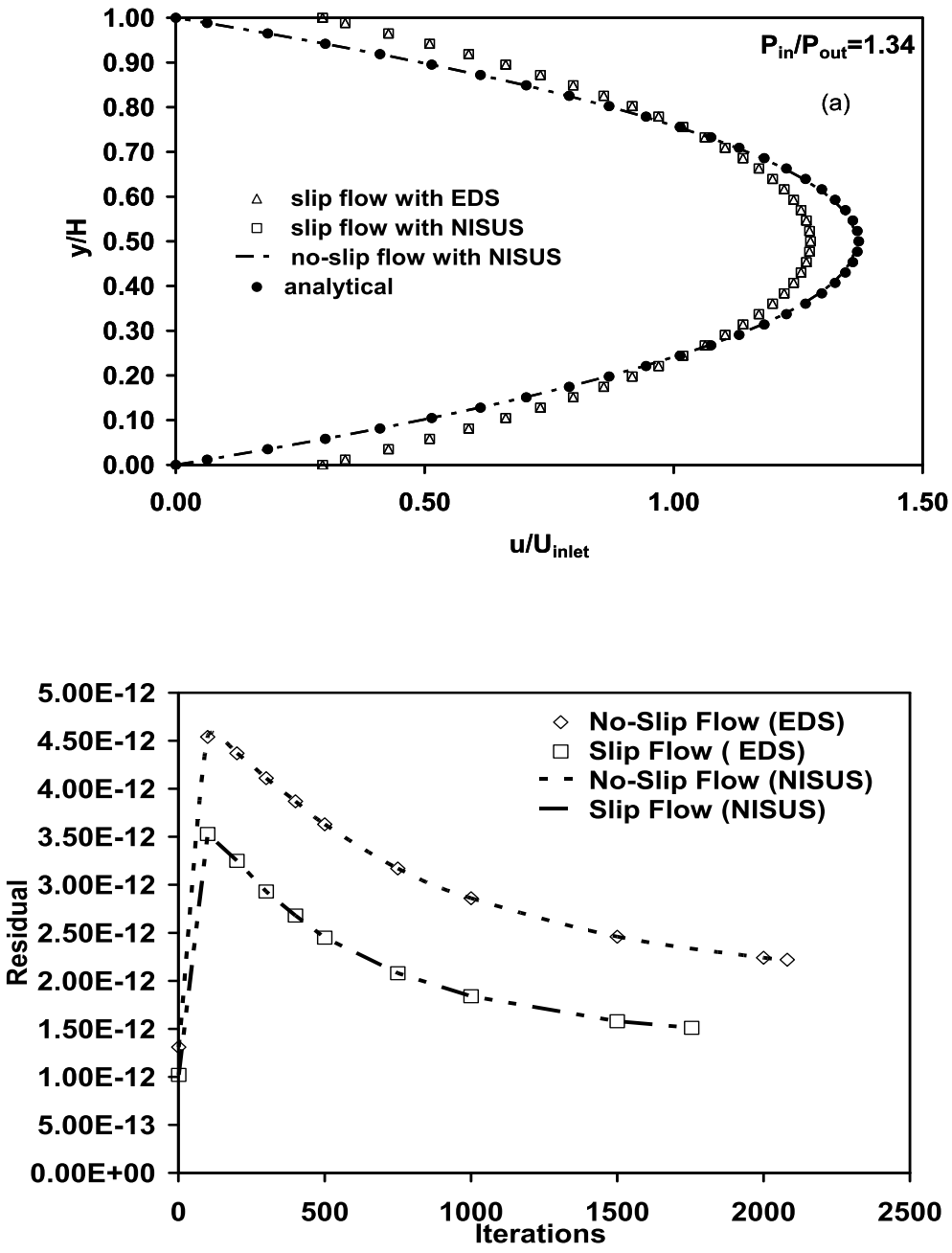


Figure 6.28: (a) Fully developed velocity profile with different schemes; (b) The residual profile with iterations before convergence

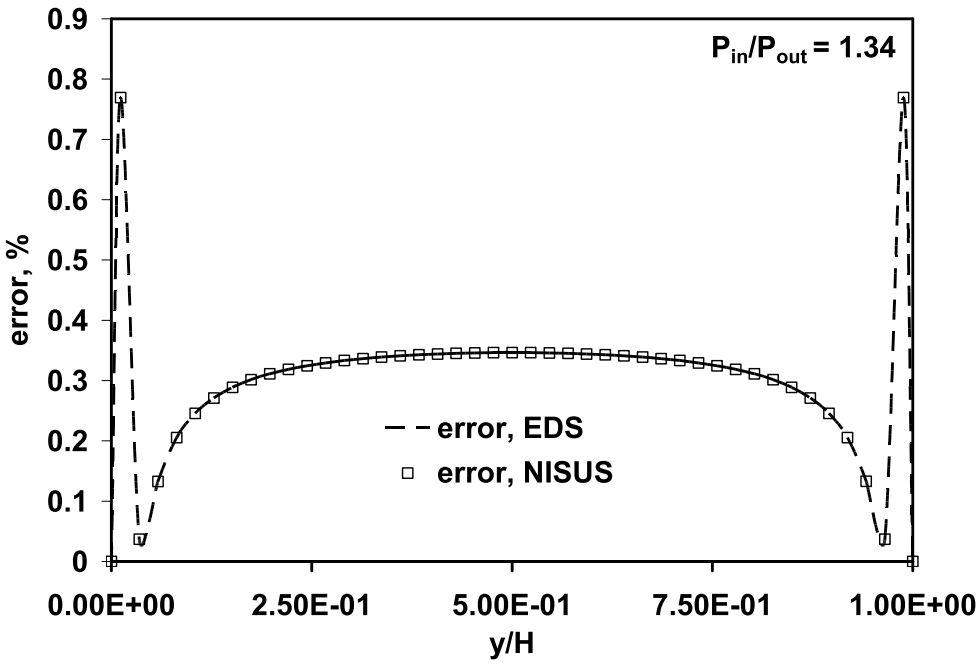


Figure 6.29: Error analysis as compared with the analytical results

about 0.31 % for both schemes. A noticeable spike in the error, as predicted by both schemes, is due to near wall effect as a result of increasing gradients of velocity and temperature solution. The profile of the entropy production within the microchannel is dependent on both the temperature and velocity fields, as obtained from Eqn. 6.6.4. The dependence on the velocity gradients is stronger, especially at lower pressure ratios. Since the streamwise velocity gradients are dominant as compared with the wall normal velocity gradients, it should be expected that the profile of the entropy generation will maintain the same pattern with the streamwise velocity profile.

$$\dot{P}_s = \frac{\mu}{T}\left\{\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2 + 2\left(\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2\right)\right\} \tag{6.6.4}$$

The predicted distribution of pressure along the microchannel is linear (as expected), since the flow is incompressible and fully developed. Predicted distributions at varying pressure ratios are shown in Fig. 6.30. A steeper pressure gradient is observed for the no-slip case, due to higher friction along the wall of the microchannel. A grid refinement study is shown in Fig. 6.31. The number of iterations required for solution convergence increases when the grid spacing is reduced (as expected). It can be observed that the trends remain similar for varying pressure ratios, although fewer iterations are needed at lower pressure ratios. The corresponding CPU run-times become lower with fewer iterations. Little difference between NISUS and EDS was observed in this problem, since the flow is uni-directional and additional upstream dependence on flow directionality does not appreciably improve solution economy or convergence.

Fig. 6.32 presents the effects of grid refinement on the total entropy production

and the number of iterations before steady state convergence. Grid independent results in term of the total entropy production are observed between the refinement indices of 4.7 and 4.9. It can be observed that further refinements beyond this range result in speedy descent in the total entropy production. However, the cost of simulation in term of higher rate of computation before convergence quickly cancels this advantage. Fig. 6.33 shows the variation of entropy production along the length of the microchannel with the corresponding influence of the velocity slip coefficients. The first slip coefficient (i.e., the coefficient of the velocity gradient term in Eqn. 6.6.1), ξ_1 , is constant at the specified accommodation coefficient of 0.8 as shown in Fig. 6.33(a). Due to linear descent in temperature distribution along the microchannel, the second slip coefficient (i.e., the coefficient of the temperature gradient term in Eqn. 6.6.1), ξ_2, increases linearly along the microchannel. This variation contributes to the non-linear profile observed for the entropy production (see Fig. 6.33(b)).

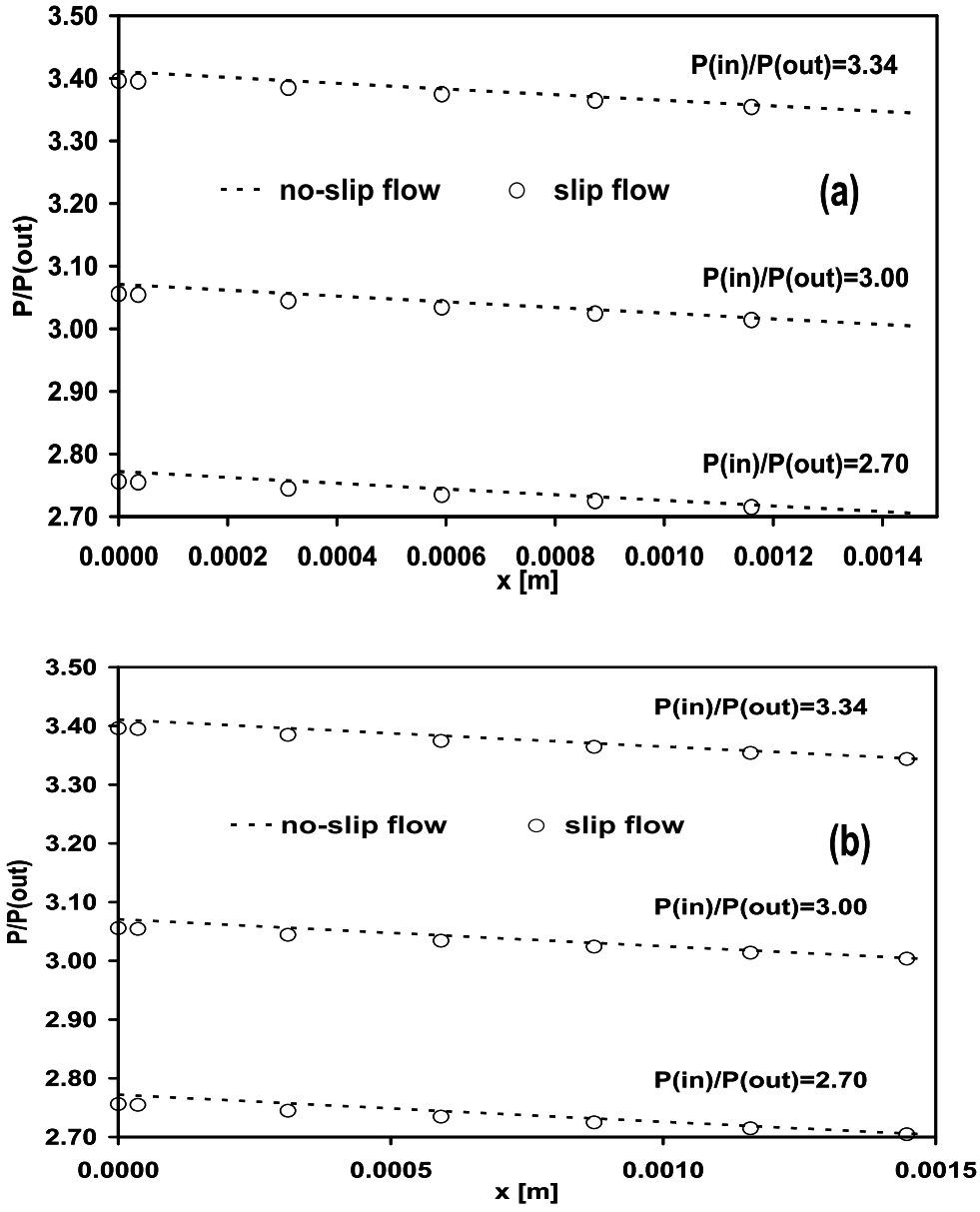


Figure 6.30: Pressure gradient at different pressure ratios; (a) conventional EDS; (b) NISUS variant of EDS

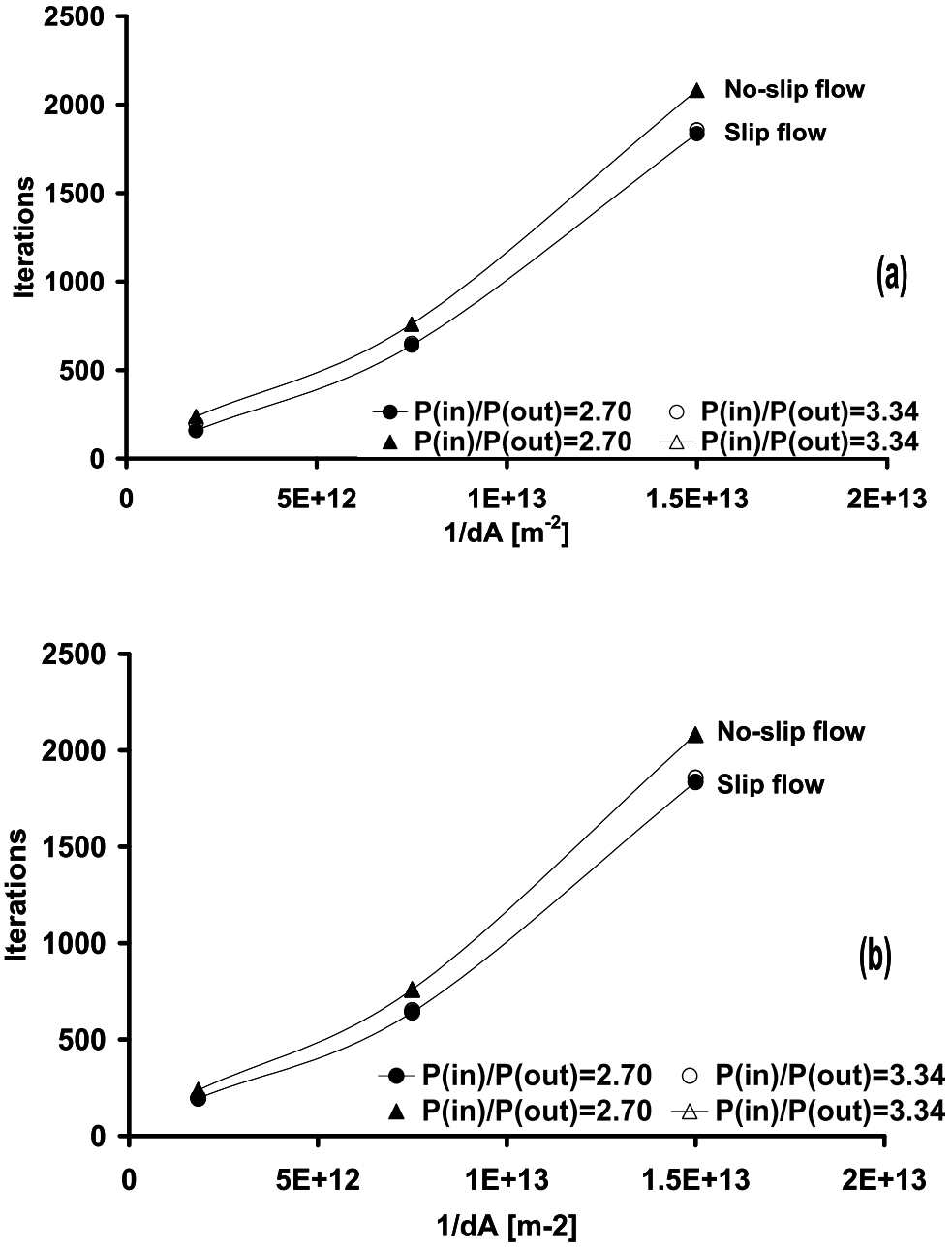


Figure 6.31: Rate of steady state convergence with refinement; (a) Conventional EDS; (b) NISUS variant of EDS

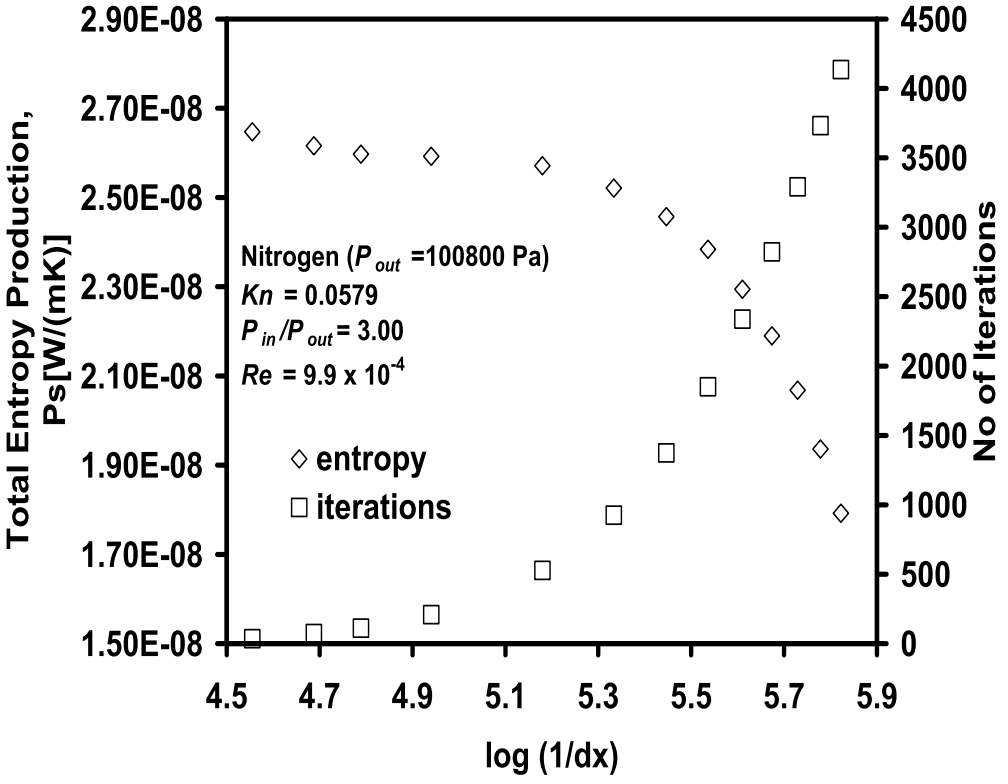


Figure 6.32: Total entropy production with refinement

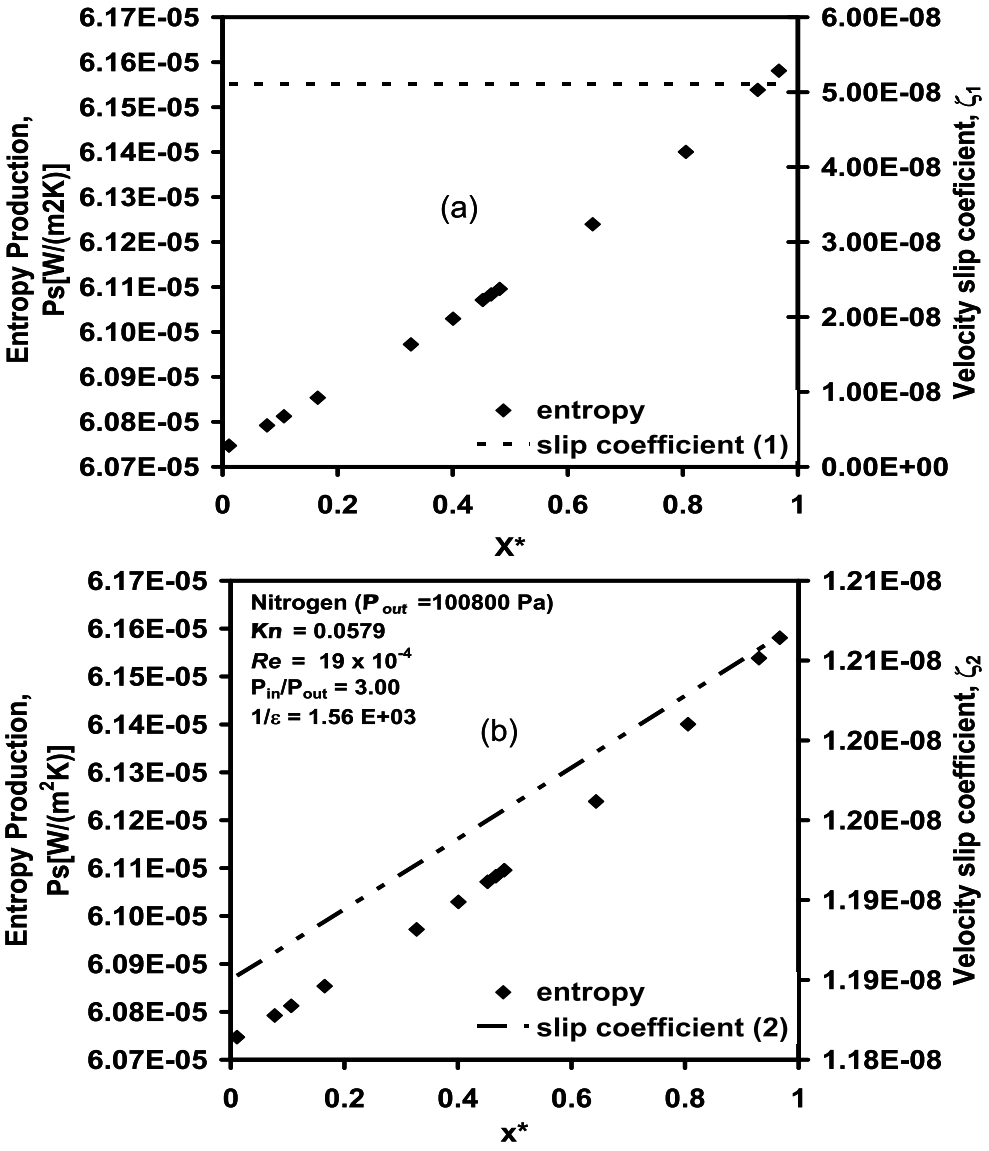


Figure 6.33: Entropy production along the microchannel; (a) with ξ_1; (b) with ξ_2

Chapter 7

Conclusions and Recommendations for Future Research

7.1 Conclusions

Three variants of a new mass-weighting convective upwinding scheme (Non-Inverted Skew Upwind Scheme, NISUS) and a new Compressed Banded Data (CBD) format have been presented in this thesis. Conclusions regarding these advances are listed below.

1. The 3-node / 3-point NISUS possesses an asymmetric distribution of active interpolation points for convective upwinding of the scalar about the desired integration point. In contrast, the 4-node / 8-point NISUS possesses a symmetric distribution of active interpolation points for convective upwinding of the scalar about the integration point. Both the 3-node / 3-point NISUS and 4-node / 8-point NISUS are developed with three-dimensional hexahedral elements and a Control Volume-Based Finite Element Method (CVFEM).

2. A NISUS variant of the Exponential Differencing Scheme has been formulated with a staggered grid finite volume method.

3. The proposed schemes are fully conservative, and they provide an explicit representation of convected integration point variables in terms of nodal values. As a result, costly inversion of local 12×12 upwind coefficient matrices is avoided in the three-dimensional CVFEM.

   The 4–node/ 8–point NISUS offers better accuracy at high Peclet numbers with circulating flows. An added advantage of this new scheme was demonstrated by a CPU time savings, while avoiding local matrix inversion of the convection upwind variables. New upwind interpolation functions may permit higher convection accuracy, as the upwind point moves closer to the integration point in the upstream differencing.

4. Also, a new storage format called Compressed Banded Data (CBD) has been developed for matrices generated by the CVFEM (Control Volume-Based Finite Element Method). The new design is capable of scanning the coefficient matrix and the right-hand vector without any internal changes to the CVFEM formulation, thereby allowing dynamic switching between different solvers, such as direct and iterative solvers. Reduced storage requirements with the CBD format have been presented, as well as performance characteristics of various iterative solvers utilizing this new format. It was observed that the ILU preconditioned GMRES (m) shows better accuracy and convergence characteristics over a wider range of heat transfer problems, as compared with CG, CGS and Bi-CGSTAB Krylov methods.

5. A Tri-Diagonal Matrix Algorithm (TDMA) with the Line Gauss-Seidel (LGS) solver has been developed and utilized with the SIMPLEC solution method.

The newly developed solvers and schemes have been applied to several engineering problems involving heat transfer and fluid flow. The CVFEM solvers are demonstrated with application to a scalar-step change problem, inlet/outlet problem and radial heat flow in a hollow sphere. The two-dimensional FVM solver was applied to additional fluid flow problems, including natural convection in a square cavity and microchannel flow. For natural convection problems, the residuals of the momentum equations decline more monotonically for NISUS than the conventional EDS. The upstream weighting on flow directionality has beneficial effects, with better stability of convective non-linearities.

7.2 Recommendations for Future Research

This thesis has demonstrated that NISUS performs well for both 3–D FEM and 2–D FVM. There are, however, further areas related to NISUS which need to be studied.

1. The solution of heat transfer problems (including radial heat transfer in a hollow sphere, step change of convective scalar, and advection in an inlet/outlet tank) was attempted with a 3-node/3-point NISUS and 4-node/8-point NISUS. Extensions for treatment of the pressure-velocity coupling, with the inclusion of the pressure gradient term in the integration point operator, are recommended. This will enable NISUS to handle three-dimensional fluid flow problems using a control volume-based finite element method. It is suggested to consider the use of a "pressure Poission model", which can be included in the integration point equation.

2. The accuracy of the NISUS variant of EDS can be further improved. A further

study of the compatibility of NISUS with the existing SIMPLEC algorithm would be useful. Since the SIMPLEC algorithm was developed with the use of relaxation factors, it is possible that the optimal relaxation for the SIMPLEC algorithm may not offer maximum benefits for NISUS. Therefore, a sensitivity study of the relaxation factor and/or modification of the algorithm may be useful.

3. It will be useful to upgrade the NISUS variant of EDS from two dimensions to three dimensions. This will enable comparisons of the performance of NISUS with the two other formulations. In this way, the best platform for NISUS in different problems will be ascertained.

4. For the applications considered in this thesis, the verification of boundary conditions for the CVFEM-based NISUS is limited to Neumman and Dirichlet conditions. The Newton boundary condition for complex engineering problems will require further development. The present manner boundary condition implementation can be improved, especially for three-dimensional problems. Improvements of boundary condition implementation may have direct benefits on the performance of NISUS, in terms of solution accuracy and economy.

5. Adaptive meshing has useful advantages in numerical analysis of complex flows. Inclusion of this procedure with NISUS can offer higher accuracy for flows with recirculation and strong wave propagation. This thesis used uniform grids for all problems. However, complex problems may require an adaptive mesh capability.

References

[1] S. Rida, F. Mckenty, F. L. Meng, M. Reggio, A staggered control volume scheme for unstructured triangular grids, International Journal for Numerical Methods in Fluids 25 (1997) 697–717.

[2] B. L. Muir, B. R. Baliga, Solution of three-dimensional convection-diffusion problems using tetrahedral elements and flow-oriented upwind interpolation functions, Numerical Heat Transfer 9 (1986) 143–162.

[3] B. R. Baliga, S. V. Patankar, A new finite-element formulation for convection-diffusion problems, Numerical Heat Transfer 3 (1980) 393–409.

[4] G. E. Schneider, M. J. Raw, A skewed, positive influence coefficient upwinding procedure for control-volume-based finite-element convection-diffusion computation, Numerical Heat Transfer 9 (1986) 1–26.

[5] R. M. Smith, A. G. Hutton, The numerical treatment of advection: A performance comparison of current methods, Numerical Heat Transfer 5 (1982) 439–461.

[6] Y. A. Hassan, J. G. Rice, J. H. Kim, A stable mass-flow-weighted two-dimensional skew upwind scheme, Numerical Heat Transfer 6 (1983) 395–408.

[7] G. D. Raithby, Skew upstrem differencing schemes for problems involving fluid flow, Computer Methods Applied Mech. Engr. 9 (1976) 153–164.

[8] E. O. B. Ogedengbe, G. F. Naterer, Convective flux dependence on upstream flow directionality in finite volume computations, Numerical Heat Transfer APaper in Review, 2006.

[9] Y. Wu, R. A. Falconer, A mass conservative 3-D numerical model for predicting solute fluxes in estuarine waters, Advances in Water Resources 23 (2000) 531–543.

[10] G. F. Naterer, Heat Transfer in Single and Multiphase Systems, CRC Press, Boca Raton, FL, 2002.

[11] G. F. Naterer, Eulerian three-phase formulation with coupled droplet flow and multiphase heat transfer, Numerical Heat Transfer B 47 (4) (2003) 331–352.

[12] G. F. Naterer, J. A. Camberos, Entropy and the second law in the numerical simulation of fluid flow and heat transfer, AIAA Journal of Thermophysics and Heat Transfer 17 (3) (2003) 360–371.

[13] S. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Company, Washington DC, 1980.

[14] G. F. Naterer, D. Rinn, Towards entropy detection of anomalous mass and momentum exchange in incompressible fluid flow, International Journal for Numerical Methods in Fluids 39 (11) (2002) 1013–1036.

[15] E. O. B. Ogedengbe, G. F. Naterer, Non-Inverted Skew Upwind Scheme for three-dimensional convective transport, Numerical Heat Transfer B 46 (2) (2004) 141–164.

[16] G. F. Naterer, G. E. Schneider, PHASES model of binary constituent solid - liquid phase transition- Part 1. Numerical Method, Numerical Heat Transfer B 28 (2) (1985) 111–126.

[17] P. Encyclopedia (Ed.), Schemes for Convection Discretization, Simuserve Ltd, CHAM, Wimbledon Village, London, UK, 1998.

[18] C. Carey, T. I. Scanlon, S. M. Fraser, SUCCA - A new scheme to reduce the effects of multidimensional false diffusion, PHOENICS Journal of CFD and Its Applications 5 (2) (1992) 134–174.

[19] M. A. R. Sharif, A. A. Busnaina, Investigation into the numerical dispersion problem of the skew upwind finite difference scheme, Applied Mathematical Modeling 12 (1988) 98–108.

[20] J. E. Fromm, A method for reducing dispersion in convective difference schemes, Journal of Computational Physics 3 (1968) 176.

[21] J. Zhu, W. Rodi, A low dispersion and bounded convection scheme, Computer Methods in Applied Mechanics and Engineering 98 (1991) 345.

[22] M. Darwish, F. Moukalled, A new approach for building bounded skew-upwind schemes, Computer Methods in Applied Mechanics and Engineering 129 (3) (1996) 221–233.

[23] P. H. Gaskell, A. K. C. Lau, Curvature-compensated convective transport: SMART, A new boundedness-preserving transport algorithm, International Journal for Numerical Methods in Fluids 8 (1988) 617.

[24] G. D. Albada, V. Albada, B. V. Leer, W. W. Roberts, A comparative study of computational methods in cosmic gas dynamics, Astronomy and Astrophysics 108 (1982) 76.

[25] G. F. Naterer, Contructing an entropy-stable upwind scheme for compressible fluid flow computations, AIAA Journal of Thermophysics and Heat Transfer 37 (3) (1999) 303–312.

[26] B. P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Computer Methods in Applied Mech. Engr. 19 (1979) 59–98.

[27] R. K. Agrawal, A third-order accurate upwind scheme for Navier-Stokes at high reynolds numbers, AIAA Journal of Thermophysics and Heat Transfer 81 (1981) 112.

[28] B. P. Leonard, M. A. Leschziner, J. McGuirk, The QUICK algorithm: A uniformly 3rd-order finite-difference method for highly convective flows, in: Proceedings of the First Conference on Numerical Methods in Laminar and Turbulent Flow, Swansea, UK, 1978, p. 807.

[29] B. R. Hutchinson, G. D. Raithby, A multigrid method based on the additive correction strategy, Numerical Heat Transfer 9 (1986) 511–537.

[30] O. B. Adeyinka, G. F. Naterer, Apparent entropy production difference with heat and fluid flow irreversibilities, Numerical Heat Transfer B 42 (5) (2002) 411–436.

[31] K. R. Kictley, Renormalization group based algebraic turbulence model for three-dimensional turbomachinery flows, AIAA Journal 30 (6) (1992) 1500–1506.

[32] E. O. B. Ogedengbe, 3-D control volume finite element modelling of an artificial ground freezing system, Internal Report, Department of Mechanical & Manufacturing Engineering, University of Manitoba (2003).

[33] E. O. B. Ogedengbe, G. F. Naterer, Three–dimensional distributed mass weighting for non-inverted convective skew upwinding, AIAA Journal of Therm. & Heat Transfer 18 (4) (2004) 502–510.

[34] K. Phoon, K. Toh, S. Chan, F. Lee, An efficient diagonal preconditioner for finite element solution of biot's consolidation equations, International Journal for Numerical Methods in Engineering 55 (4) (2002) 377–400.

[35] A. Mazzia, G. Pini, Numerical performance of preconditioning techniques for the solution of complex sparse linear systems, Communications in Numerical Methods in Engineering 19 (2003) 37–48.

[36] A. J. Chorin, Numerical solution of the navier-stokes equations, Mathematics of Computation 22 (1968) 745–762.

[37] W. B. Tsai, W. W. Lin, C. C. Chieng, Convergence acceleration by self-adjusted time-step size using bi-cgstab method for turbulent separated flow computation, International Journal for Numerical Methods in Fluids 39 (2002) 141–159.

[38] W. B. Tsai, W. W. Lin, C. C. Chieng, Convergence acceleration by varying time-step size using bi-cgstab method for turbulent flow computation, Numerical Methods in Partial Differential Equations 17 (2001) 454–474.

[39] E. O. B. Ogedengbe, G. F. Naterer, Dissipative kinetic and thermal energy exchange in microchannel flows, in: CSME Forum 2006, Paper 2006-77, CSME, Calgary, Alberta, 2006.

[40] R. F. Handy, D. J. Silvester, J. W. Chew, A comparison of coupled and segregated iterative solution techniques for incompressible swirling flow, International Journal for Numerical Methods in Fluids 22 (1996) 353–373.

[41] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd Edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[42] E. Montagne, A. Ekambaram, Optimal storage format for sparse matrices, Information Processing Letters 90 (2004) 87–92.

[43] R. Barrett, M. Berry, T. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, H. van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.

[44] I. Catton, Natural convection in enclosures, in: Proceedings of the 6^{th} International Heat Transfer Conference, Vol. 6, Toronto, Canada, 1978.

[45] C. Cai, I. Boyd, J. Fan, Direct simulation methods for low-speed microchannel flows, Journal Thermophysics and Heat Transfer 14 (3) (2000) 368–378.

[46] M. Wang, Z. Li, Simulations for gas flows in microgeometries using the direct simulation monte carlo method, International Journal of Heat and Fluid Flow 25 (2004) 975–985.

[47] R. Prud'homme, T. Chapman, J. Bowen, Laminar compressible flow in a tube, Applied Science Resources 43 (1986) 67–74.

[48] J. Harley, Y. Huang, H. Bau, J. Zemel, Gas flow in micro-channels, Journal Fluid Mechanics 284 (1995) 257–274.

[49] E. B. Arkilic, M. A. Schmidt, K. S. Breuer, Gaseous slip flow in long microchannels, J. Microelectromechanical Systems 6 (2) (1997) 167–178.

[50] K. Pong, C. Ho, J. Liu, Y. Tai, Non-linear pressure distribution in uniform microchannels, in: Applied Microfabrication to Fluid Mechanics, ASME Winter Annual Meeting, Chicago, IL, 1994, pp. 51–56.

[51] A. Beskok, G. Karniadakis, Simulation of heat and momentum transfer in complex microgeometries, Journal of Thermophysics and Heat Transfer 8 (1994) 647–655.

[52] E. Piekos, K. Breuer, DSMC modeling of micromechanical devices, Paper 95-2089, AIAA Thermophysics Conference, San Diego, CA, 1995.

[53] S. V. Patankar, D. B. Spalding, A finite–difference procedure for solving the equations of the two-dimensional boundary layers, International Journal of Heat and Mass Transfer 10 (1967) 1389–1412.

[54] O. C. Zienkiewicz, R. Loehner, K. Morgan, S. Nakazawa, Finite Elements in Fluids, Vol. 5 of Finite Elements in Fluid Mechanics–A Decade of Progress, Wiley, New York, 1984.

[55] G. E. Schneider, Elliptic Systems: Finite-Element Method I, Handbook of Numerical Heat Transfer, John Wiley & Sons, Inc., 1988.

[56] E. O. B. Ogedengbe, G. F. Naterer, Reduced design time with NISUS convection modelling: Application to a rotating lubrication system, in: 7th Africa-USA International Conference on Manufacturing Technology, Port Harcourt, Nigeria, 2004, pp. 14–20.

[57] E. O. B. Ogedengbe, G. F. Naterer, Preconditioned solver performance with compressed banded data format in 3-D convective heat transfer simulations, Numerical Heat Transfer A 48 (10) (2005) 965–985.

[58] G. E. Schneider, M. Zedan, Control volume based finite element formulation of the heat conduction equation, in: AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, AIAA, AIAA, St. Louis, Missouri, 1982.

[59] K. H. Huebner, E. A. Thornton, The Finite Element Method for Engineers, 2nd Edition, John Wiley & Sons, 1982.

[60] M. J. Raw, A new control-volume-based finite element procedure for the numerical solution of the fluid flow and scalar transport equations, Phd thesis, University of Waterloo, Waterloo, Ontario (1985).

[61] G. D. Raithby, G. Schneider, Numerical solution of problems in incompressible fluid flow: Treatment of the velocity-pressure coupling, Numerical Heat Transfer 7 (1979) 417–440.

[62] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.

[63] J. P. V. Doormaal, G. D. Raithby, Enhancements of the SIMPLE for predicting incompressible fluid flows, Numerical Heat Transfer 7 (1984) 147–163.

[64] C. M. Rhie, W. L. Chow, Numerical study of the turbulent flow past and airfoil with trailing edge separation, AIAA Journal 21 (1983) 1525–1532.

[65] J. P. V. Doormaal, G. D. Raithby, Enhancements of the simple method for predicting incompressible fluid flows, Numerical Heat Transfer 7 (1984) 147–163.

[66] H. A. V. der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems, SIAM Journal of Scientific and Statistical Computing 13 (1992) 631–644.

[67] H. Wolkowicz, Solving semi-definite programs using preconditioned conjugate gradients, Optimization Methods and Software 19 (6) (2004) 653–672.

[68] N. H. M. Ali, D. J. Evans, Preconditioned rotated iterative methods in the solution of elliptic partial differential equations, International of Computer Mathematics 81 (9) (2004) 1163–1174.

[69] M. D. Hughes, K. Chen, An efficient preconditioned iterative solver for solving a coupled fluid structure interaction problem, International Journal of Computer Mathematics 81 (5) (2004) 583–594.

[70] E. O. B. Ogedengbe, G. F. Naterer, Compressed banded data structure for preconditioned iterative solver in numerical heat transfer, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, Paper 2005-0572, AIAA, Reno, Nevada, 2005.

[71] Y. S. Nam, H. G. Choi, J. Y. Yoo, AILU preconditioning for the finite element formulation of the incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering 191 (2002) 4323–4339.

[72] M. Heniche, Y. Secretan, M. Leclerc, Efficient ILU preconditioning and inexact Newton-GMRES to solve the 2d steady shallow water equations, Communications in Numerical Methods in Engineering 17 (2001) 69–75.

[73] J. K. Reid, The use of conjugate gradients for systems of equations possessing property a, SIAM Journal of Numerical Analysis (1972) 325–332.

[74] P. Sonneveld, CGS - A fast lanczos-type solver for non-symmetric linear systems, SIAM Journal on Scientific and Statistical Computing 10 (1989) 36–52.

[75] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856–869.

[76] H. A. V. der Vorst, Iterative Krylov Methods for Large Linear Systems, Cambridge University Press, New York, 2003.

[77] C. Prakash, Examination of the upwind (Donor-Cell) formulation in control volume finite-element methods for fluid flow and heat transfer, Numerical Heat Transfer 11 (1987) 401–416.

[78] W. Schonauer, K. Raith, G. Glotz, The SLDGL program package for the self-adaptive solution of non-linear system of elliptic and parabolic PDE's, Proc. Fourth IMACS International Symposium Computer Methods for PDE's, Lehigh University, Lehigh, Pa.

[79] W. Schonauer, K. Raith, G. Glotz, The self-adaptive solution of non-linear 2-D boundary value problems in rectangular domain, in: Proceeding International Conference Numerical Methods in Laminar and Turbulent Flow, Venice, Pineridge, 1981.

[80] E. O. B. Ogedengbe, G. F. Naterer, Distributed mass weighting of non-inverted convection upwind variables for three-dimensional flow simulations, in: 42nd

AIAA Aerospace Sciences Meeting and Exhibit, Paper 2004-995, AIAA, Reno, Nevada, 2004.

[81] G. Howell, Parallel Algorithms and Scientific Computing, High Performance and Grid Computing, Information Technology Division, North Carolina State University, Raleigh, NC (2004).

[82] G. D. V. Davis, Natural convection of air in a square cavity: A bench mark numerical solution, International Journal for Numerical Methods in Fluids 3 (1983) 249–264.

[83] S. Roy, R. Raju, H. F. Chuang, B. A. Cruden, M. Meyyappan, Modelling gas flow through microchannels and nanopores, Journal of Applied Physics 93 (2003) 4870–4879.

[84] H. Xue, Q. Fan, A new analytic solution of the navier-stokes equations for microchannel flows, Microscale Thermophysical Engineering 4 (2000) 125–143.