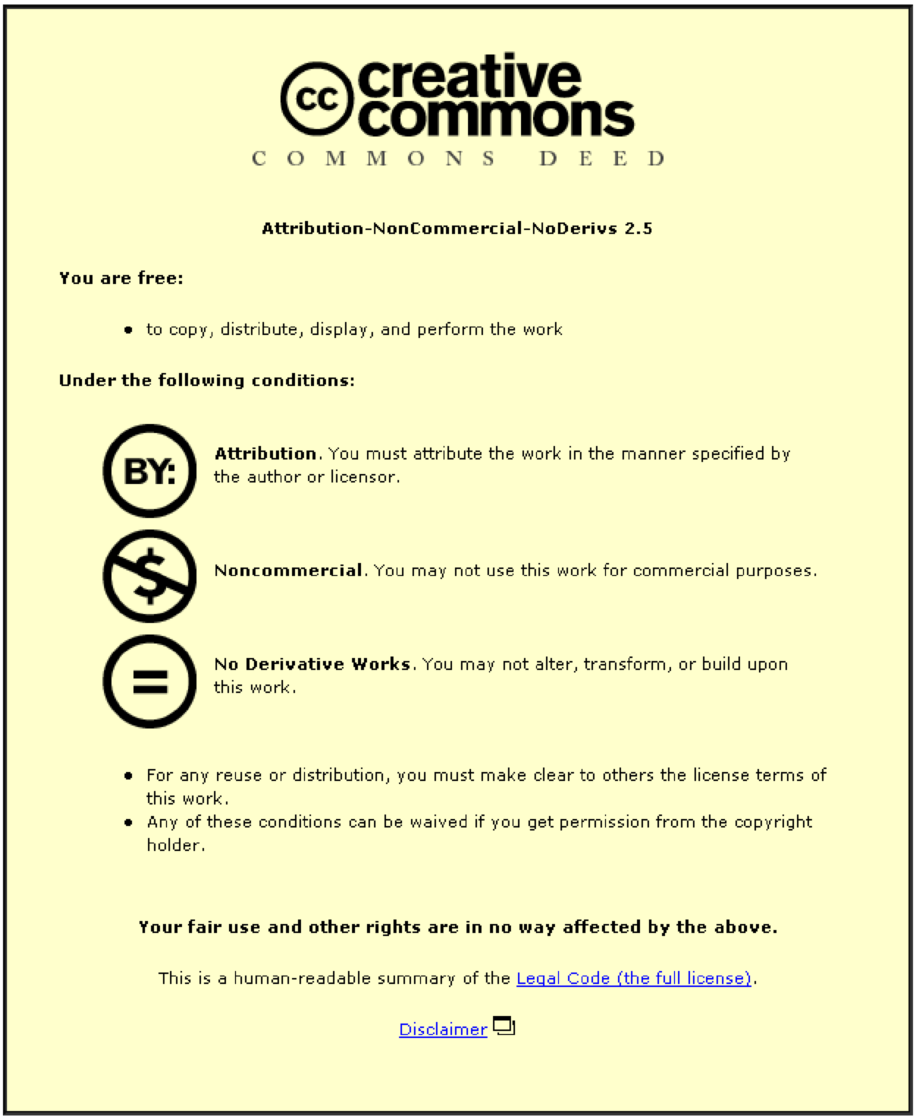


Three Dimensional Finite Element Modelling of Non-Newtonian Fluid Flow Through a Wire Mesh

Kee Chien TING

2005

A doctoral thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy degree of Loughborough University.

Advanced Separation Technologies Group
Department of Chemical Engineering
Loughborough University

ACKNOWLEDGEMENT

I wish to express my deepest gratitude to supervisor Professor Vahid Nassehi and co-supervisor Professor Richard J. Wakeman of Chemical Engineering Department, Loughborough University for their invaluable advice and patient guidance in my study. Their expert opinion, input and comments have consistently inspired and challenged me throughout the duration of my research with the department.

The unceasing support and encouragement given by my wife and my family have been a constant motivation for me to strife ahead in my research towards the completion of my PhD research.

ABSTRACT

Monofilament cloths are used as the separation media in filtration; woven wire cloths or screens are also used as the media in filters or to enhance the integrity of the filter medium in, for example, filter cartridges. A better understanding of the flow pattern in the woven structure is essential in examining the initial stages of cake filtration as well as the effect of weaves on fouling phenomena within a filter cloth.

Due to the complex geometry of a woven cloth, three-dimensional modelling is necessary to correctly visualize the structure of the flow and hence to predict pressure losses. The modelling in a three-dimensional domain was handled using a finite element method which is known to cope with flow domains in complex geometries very effectively. The governing equations of continuity and momentum were solved by a mixed U-V-W-P finite element method and in conjunction with a first order Taylor-Galerkin scheme for temporal discretization. A secondary solution scheme based on a continuous Penalty finite element method in conjunction with theta time stepping method was also used to solve the governing equations.

Two robust and reliable computer tools based on these sound and robust numerical techniques have been developed to simulate Newtonian and non-Newtonian fluid flow through a woven wire mesh. Purpose-designed test cases were used to validate the capability of the developed algorithms and were found to give expected numerical predictions.

A selection of domains was used to investigate the effect of weave pattern, aperture to diameter ratio and Reynolds number on flow pattern and pressure drop. Based on these domains, simulations were successfully conducted to investigate fluid flow through four basic pore types in a plain weave, twill weave and satin weave. The flow fields in the interstices were illustrated using a commercial graphics software package. The results showed that the weave pattern has a profound effect on the fluid flow pattern and pressure drop across the wire mesh. Simulation results showed that plain weave gives the lowest pressure drop, while satin weave gives the highest pressure drop across the woven cloths.

Fluid flow through a plain weave was further investigated in conjunction with the experimental studies of Rushton (1969) using water and Chhabra and Richardson (1985) using shear-thinning fluids. Simulations were tested against experimental data extracted from both studies. The close agreement of the results to those of the available experimental data in literature showed the accuracy and the reliability of the predictions.

Personal communication with industrial experts and woven cloth manufacturers have confirmed industrial practice, whereby a plain weave is primarily used due to its lowest flow resistance. This showed that the developed model is capable of generating accurate results for flow of both Newtonian and non-Newtonian fluids through filter media. The model can be used by design engineers as a convenient and effective Computer Aided Design (CAD) tool for quantifying effects of pressure drop. The model can also be extended to describe particle capture on/in the wire mesh and woven filter cloths.

LIST OF FIGURES

Page

Chapter 2

Figure 2.1 Basic configuration of three basic weaves: (a) plain weave (b) twill weave and (c) satin weave 8

Figure 2.2 Other popular weave pattern : (a) Plain Dutch Weave (b) Dutch Twilled Weave and (c) Reversed Plain Dutch Weave 10

Figure 2.3 A magnified view of a plain weave type wire mesh 11

Figure 2.4 Four types of basic weave patterns of a wire mesh (Backer, 1951) 12

Figure 2.5 A schematic diagram of fluid flow through an orifice 14

Figure 2.6 Schematic diagrams showing (a) a column packed with spheres (b) a 'capillary bundle' model for the packed column in (a) 16

Figure 2.7 A schematic diagram showing flow past a submerged cylinder bounded by Two parallel plates 18

Figure 2.8 Flow curves of Newtonian and time-independent fluids 21

Figure 2.9 Drag coefficient versus Reynolds number for non-Newtonian fluids (Chhabra and Richardson, 1985) 29

Chapter 3

Figure 3.1 Slip at a solid wall 42

Chapter 4

Figure 4.1 The linear prism element 55

Figure 4.2 Time stepping scheme 86

Figure 4.3 The developed algorithm for U-V-W-P scheme 92

Figure 4.4 The developed algorithm for continuous Penalty scheme 93

Chapter 5

Figure 5.1 (a) Schematic diagram (b) frontal view and (c) top view of pore type 1 with aperture to diameter ratio 1.5 96

Figure 5.2 (a) Schematic diagram (b) frontal view and (c) top view of pore type 2 with aperture to diameter ratio 1.5 97

Figure 5.3 (a) Schematic diagram (b) frontal view and (c) top view of pore type 3 with aperture to diameter ratio 1.5 98

Figure 5.4 (a) Schematic diagram (b) frontal view and (c) top view of pore type 4 with aperture to diameter ratio 1.5 99

Figure 5.5 (a) Schematic diagram (b) frontal view and (c) top view of pore type 1 with aperture to diameter ratio 2.0 100

Figure 5.6 (a) Schematic diagram (b) frontal view and (c) top view of pore type 2 with aperture to diameter ratio 2.0 101

Figure 5.7 (a) Schematic diagram (b) frontal view and (c) top view of pore type 3 with aperture to diameter ratio 2.0 102

Figure 5.8 (a) Schematic diagram (b) frontal view and (c) top view of pore type 4 with aperture to diameter ratio 2.0 103

Figure 5.9 (a) Schematic diagram (b) frontal view and (c) top view of pore type 4 with aperture to diameter ratio 1.0 104

Figure 5.10 Discretized finite element mesh of domain 1 105

Figure 5.11 Cartesian vector representing *x*-, *y*- and *z*-components of velocity field 108

Figure 5.12 The colour legend bar used to represent magnitude of velocity vectors in Surfer 8 graphics software package 108

Figure 5.13 A schematic diagram of *Test Case 1* 111

Figure 5.14 The velocity vectors of *Test Case 1* in the *x*-z plane at $y = l_w/2$ for non-Newtonian fluid 112

Figure 5.15 A schematic diagram of *Test Case 2* 113

Figure 5.16 The velocity vectors of *Test Case 2* in the (a) *x*-*z* plane at $y = l_w/2$ (b) *y*-*z* plane at $x = l_l/2$ for non-Newtonian fluid 114

Figure 5.17 A schematic diagram of *Test Case 3* 116

Figure 5.18 The velocity vectors of *Test Case 3* in the (a) *x*-*z* plane at $y = l_w/2$ (b) *y*-*z* plane at $x = l_l/2$ for non-Newtonian fluid 117

Figure 5.19 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 1 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$. 121

Figure 5.20 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 1 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 122

Figure 5.21 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 1 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 123

Figure 5.22 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 1 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 124

Figure 5.23 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 1 in the *x-z* plane at y = $l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 125

Figure 5.24 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 1 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 126

Figure 5.25 Drag coefficient vs Reynolds number for Newtonian and non-Newtonian fluid flow through domain 1 127

Figure 5.26 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 2 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 129

Figure 5.27 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 2 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 130

Figure 5.28 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 2 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 131

Figure 5.29 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 2 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 132

Figure 5.30 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 2 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 133

Figure 5.31 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 2 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 134

Figure 5.32 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 2 135

Figure 5.33 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 3 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 137

Figure 5.34 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 3 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 138

Figure 5.35 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 3 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 139

Figure 5.36 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 3 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 140

Figure 5.37 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 3 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 141

Figure 5.38 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 3 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 142

Figure 5.39 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 3 143

Figure 5.40 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 4 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 145

Figure 5.41 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 4 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 146

Figure 5.42 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 4 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 147

Figure 5.43 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 4 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 148

Figure 5.44 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 4 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 149

Figure 5.45 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 4 in the *y-z* plane at $x = l_l/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 150

Figure 5.46 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 4 151

Figure 5.47 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 5 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 153

Figure 5.48 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 5 in the *y-z* plane at $x = l_l/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 154

Figure 5.49 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 5 155

Figure 5.50 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domain 1 and domain 5 156

Figure 5.51 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 6 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 158

Figure 5.52 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 6 in the *y-z* plane at $x = l_l/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 159

Figure 5.53 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 6 160

Figure 5.54 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domain 2 and domain 6 161

Figure 5.55 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 7 in the *x-z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 163

Figure 5.56 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 7 in the *y-z* plane at $x = l_l/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 164

Figure 5.57 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 7 165

Figure 5.58 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domain 3 and domain 7 166

Figure 5.59 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 8 in the *x*-*z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 168

Figure 5.60 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 8 in the *y*-*z* plane at $x = l_l/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 169

Figure 5.61 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 8 170

Figure 5.62 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domain 4 and domain 8 171

Figure 5.63 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 9 in the *x*-*z* plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 173

Figure 5.64 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 9 in the *y*-*z* plane at $x = l_l/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 174

Figure 5.65 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 9 175

Figure 5.66 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domains 4, 8 and 9 176

Figure 5.67 Drag Coefficient vs Reynolds Number for Newtonian and non-Newtonian liquid flows through domain 1 179

Figure 5.68 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domains 1, 2, 3 and 4 180

Figure 5.69 Drag Coefficient vs Reynolds Number for shear thickening fluid flow through domains 1, 2, 3 and 4 181

Figure 5.70 Drag Coefficient vs Reynolds Number for shear thinning fluid flow through domains 1, 2, 3 and 4 182

Figure 5.71 Drag Coefficient vs Reynolds Number for Newtonian fluid flow through domains 5, 6, 7 and 8 183

Figure 5.72 Drag Coefficient vs Reynolds Number for shear thickening fluid flow through domains 5, 6, 7 and 8 184

Figure 5.73 Drag Coefficient vs Reynolds Number for shear thinning fluid flow through domains 5, 6, 7 and 8 185

Figure 5.74 The y- velocity component of Newtonian fluid flow through domain 4 in the x-z plane at (a) $y = l_w$ and (b) $y = 0$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$ 188

NOMENCLATURE

A	Cross sectional area
C_D	Dimensionless discharge coefficient
c	Speed of sound in the fluid
d	Diameter
d_h	Hydraulic diameter
D_o	Orifice diameter
D'	Effective diameter
dp	Dynamic pressure difference across the media
dz	Thickness of porous media
d_p	Particle diameter
e	Element number
ec	End count
f	Friction factor
g	Gravitational acceleration
k	Permeability
L	Height of bed
l	Characteristic flow domain dimension
l_f	Domain length
l_w	Domain width
N_i	Interpolation function (elemental shape function)
n	Flow behaviour index in Power Law model
$\hat{n}_x$	Unit vector in the x- direction
$\hat{n}_y$	Unit vector in the y- direction

$\hat{n}_z$	Unit vector in the z- direction
p	Pressure
Δp	Pressure drop
pc	Pick count
Q	Volumetric flow rate
Re	Reynolds number
Re_{NN}	Non-Newtonian fluid Reynolds number
S_v	Boundaries where Dirichlet type boundary conditions were specified
S_f	Boundaries where Neumann type boundary conditions were specified
S	Collective boundaries of the entire computational domain
T	Temperature
t	Time
$\hat{t}$	Unit vector tangent to boundary
u	Superficial velocity
u_o	Velocity through orifice
v_b	Velocity of the solid wall
v_x	Velocity in x-direction
v_y	Velocity in y-direction
v_z	Velocity in z-direction
W	Wetted orifice parameter
W_i	Weighted function

Greek Symbols

α	Time increment parameter in Taylor series of expansion
β	Slip coefficient

β_0	Initial slip coefficient
Γ	Boundary of the solution domain
Γ_e	Boundary of an element domain
δ	Kronecker delta
ε	Porosity
ϕ	Null set
$\dot{\gamma}$	Strain rate
η	Apparent Viscosity
η_0	Consistency Index used in Power Law equation
λ	Penalty parameter
μ	Fluid viscosity
θ	Time stepping parameter used in the implicit theta method
ρ	Density
σ	Cauchy stress tensor
τ	Stress tensor
Ω	Solution domain
Ω_e	An element domain within the solution domain

Other Symbols

∇	Operator nabla ($\equiv i\frac{\partial}{\partial x}+j\frac{\partial}{\partial y}+k\frac{\partial}{\partial z}$)
∇^2	Operator Laplacian ($= \nabla\cdot\nabla = \frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}$)

TABLE OF CONTENTS

Page

Certificate of originality i

Acknowledgement ii

Abstract iii

List of Figures v

Nomenclature xii

1 INTRODUCTION 1

1.1 Significance of The Topic of Study 1

1.2 Objectives of The Study 4

1.3 Structure of The Thesis 5

2 LITERATURE REVIEW 7

2.1 Weave Patterns of Wire Meshes 7

2.1.1 Characteristics of Various Weave Patterns 7

2.1.2 Resulting Pore Types 12

2.2 Relevant Models Describing Flow Through a Wire Mesh 13

2.2.1 Orifice Analogy 13

2.2.2 Random Packed Bed Analogy 15

2.2.3 Flow Over Cylinder Analogy 17

2.3 Non-Newtonian Fluid Flow Through A Wire Mesh 18

2.3.1 Properties of Non-Newtonian Liquid 19

2.3.2 Experimental Studies 21

2.3.3 Numerical Studies 32

2.4 Closure 34

3 GOVERNING EQUATIONS AND BOUNDARY CONDITIONS 35

3.1 Flow Model 35

3.1.1 Continuity Equation 35

3.1.2 Momentum Equation 37
3.2 Constitutive Equation 39
3.3 Assumptions 40
3.3.1 Assumptions Relating to Wire Mesh Geometry 40
3.3.2 Wall Slip 41
3.4 Boundary Conditions 43
3.4.1 Inlet Boundary Condition 45
3.4.2 Outlet Boundary Condition 46
3.5 Closure 46

4 DERIVATION OF THE WORKING EQUATIONS 47
4.1 Mathematical Modelling Strategies 47
4.2 Finite Element Method 50
4.3 Choice of Interpolation Functions 53
4.4 Inf-Sup Condition 56
4.5 Solution Schemes of Stokes Equation 61
4.5.1 Perturbed Continuity Equation 61
4.5.2 Continuous Penalty Method 78
4.5.2.1 Calculation of Pressure 84
4.5.2.2 Solution of the Transient Equations 84
4.6 Convergence of the Solution 89
4.7 Mesh Refinement 90
4.8 Output 91
4.9 Developed Algorithm 91
4.10 Closure 94

5 RESULTS AND DISCUSSIONS 95
5.1 Domain Discretization and Boundary Conditions 95
5.1.1 Domains Studied and Boundary Conditions 95
5.1.2 Physical Properties 107
5.2 Presentation of Results 107

5.3 Validation of Model 109

5.3.1 Test Case 1: Rectangular Domain 110

5.3.2 Test Case 2: Rectangular Domain with Two Cut Sized Cylinders 112

5.3.3 Test Case 3: Rectangular Domain with Four Cut Sized Cylinders 115

5.3.4 Mesh Refinement 118

5.4 Results from Domain 1: Pore Type 1 with Aperture/Diameter Ratio of 1.5 119

5.5 Results from Domain 2: Pore Type 2 with Aperture/Diameter Ratio of 1.5 128

5.6 Results from Domain 3: Pore Type 3 with Aperture/Diameter Ratio of 1.5 136

5.7 Results from Domain 4: Pore Type 4 with Aperture/Diameter Ratio of 1.5 144

5.8 Results from Domain 5: Pore Type 1 with Aperture/Diameter Ratio of 2.0 152

5.9 Results from Domain 6: Pore Type 2 with Aperture/Diameter Ratio of 2.0 157

5.10 Results from Domain 7: Pore Type 3 with Aperture/Diameter Ratio of 2.0 162

5.11 Results from Domain 8: Pore Type 4 with Aperture/Diameter Ratio of 2.0 167

5.12 Results from Domain 9: Pore Type 4 with Aperture/Diameter Ratio of 1.0 172

5.13 Pressure Drop Analysis 177

5.14 Closure 189

6 CONCLUSIONS AND RECOMMENDATIONS 190

6.1 Conclusions 190

6.2 Recommendations for Future Work 192

REFERENCES

APPENDICES

A1 Relevant Mathematical Bookwork

A2 Publications

A3 Program Manual

A4 Program Listing

A5 Sample Input File

A6 Sample Output File

CHAPTER 1

INTRODUCTION

1.1 Significance of the Topic of Study

Woven screens have been widely used in the construction of conventional sieves, filters and separators in solid-liquid processes in the filtration of polymers, chemicals, pharmaceutical, cosmetic, hydraulic oil and fuels, as well as the luxury food and beverage industries. In these applications, woven wire meshes are used either as the support for finer filter media, for example in filter cartridges, or as the filtering medium itself.

Recently, advances in weaving technology have led to increasing applications of single-layered and laminated multi-layered composite wire screens of various metals such as stainless steel, titanium, alloy, non-ferrous metals as well as novel composites materials. These screens can be customized into various shapes and forms such as filter candles, circles and discs depending on applications and design. Metal screens, which are characterized by chemical and heat resistance against corrosion and abrasion, extremely high mechanical stability, high durability, wear resistant, efficient retention of solids, and consistent flow rates during the filtration cycle have drawn increasing interest as a replacement for conventional cloth filters since they are reusable, recyclable and are therefore environmentally friendly.

Wire mesh is also widely used in the aerospace industry, particularly in the fields of lightning strike protection mesh for composites, acoustics meshes applied to the housing of the jet engines for noise reduction, meshes for electromagnetic interference (EMI) shielding application of displays, anti-turbulence-screens in wind tunnels and fuel and hydraulic systems micronic filtration meshes. Remarkable advances in manufacturing skills in the recent decade have also seen ring shape wire meshes produced in small sizes for use as distillation columns packing.

With aperture fineness ranging from 1 μm up to 10 mm for processes involving fluids of diverse properties and suspensions, the performance of these woven wire meshes should be reliably predicted to enable engineers to design and evaluate applications across industries. An understanding of fluid flow through these screens is therefore vital.

Despite the significance and numerous applications in industry, studies on woven materials have been limited and in these studies attention has focused on permeability, porosity and resistance properties. Investigations of flow of non-Newtonian liquids through various single screens are relatively scarce and mostly concentrated on shear thinning fluids. No researchers have previously studied the velocity and pressure profile as fluid flows past the wire mesh interstices. A better understanding of the flow characteristics in the interstices of the woven structure could prove useful in examining the initial stages of cake filtration as well as the effect of weaves on fouling phenomena within a filter cloth.

Considering the significance of knowledge in the flow behaviour of non-Newtonian fluids, it is necessary to research the complex flow field generated by the flow of non-Newtonian fluid through a complex geometry. Hence, the significance of studying the flow of a non-Newtonian fluid through a wire mesh cannot be overstressed.

Computational fluid dynamics (CFD) techniques have been known to provide a powerful and convenient route for the qualitative and quantitative analysis of non-Newtonian flow systems. Computer simulations offer the opportunity of studying important steady-state and transient flow free from instrumental and environmental effects. With increasingly powerful processors and more economical computing costs, accurate computations of very complex flow problems have been made possible in two-dimensional and even three-dimensional domains. A great number of computational researches have been undertaken and published in the field of CFD. Therefore, CFD seems to be a feasible alternative to study flow through weave structures that are characterized by small apertures in the mm and μm size ranges.

There are several categories of numerical techniques used in the CFD studies, of which *Finite Element Method* (FEM), has been found to be a powerful technique for solving engineering flow problems with geometrical complex domains. For this reason, the present study of complex three-dimensional flow domain is based on the FEM.

1.2 Objective of the Study

The primary objective of the present work is to develop a numerical model for predicting non-Newtonian fluid flow through a wire mesh. Due to the complex geometry of a wire mesh, three-dimensional modelling is necessary to visualize correctly the structure of the flow and hence to predict pressure losses.

This study will set a major milestone towards studying the mechanics involved in the clogging by particles in the woven media interstices during the formation of a filtration cake. It is hoped that the outcome of this study will enhance the Computer Aided Design (CAD) of high performance woven fabrics and filters.

The stages involve in achieving the described objectives are as follows:

(i) Development of a user-friendly computer code to solve Newtonian and non-Newtonian fluid flow in three-dimensional domains.

(ii) Checking the computer code using appropriate test cases on Newtonian and non-Newtonian fluids.

(iii) Selection of appropriate boundary conditions for simulating the described flow problem.

(iv) Conducting numerical simulations on flow of Newtonian fluid and shear thinning and shear thickening non-Newtonian fluids through a wire mesh.

(v) Conducting pressure drop analysis on the results.

1.3 Structure of the Thesis

This thesis consists of six sections with additional sections for references and appendices. The contents of each section are listed as follows:

1. Introduction

The subject of the present study and the significance of the present work are discussed. The work involved in achieving the overall objectives of this study are clearly outlined.

2. Literature Review

This section contains a review of the studies carried out by various researchers in fluid flow through a monofilament woven wire mesh. Various approaches used by researchers in describing flow through weave patterns are discussed and the relevant results are also presented.

3. Governing Equations and Boundary Conditions

The governing continuity and motion equations used in this study are explained. The constitutive equation used to describe non-Newtonian rheology is briefly discussed. The assumptions relating to the numerical study of the wire mesh are outlined and the boundary conditions prescribed in this study are explained.

4. Derivation of the Working Equations

Following the discussion of the governing equations in Chapter 3, the numerical technique used in this study is explained. The solution schemes of the equations are followed through in two numerical methods. The final working equations for each solution scheme are presented. The strategies for the algorithm design of each solution scheme are also outlined.

5. Results and Discussions

Purpose-designed test cases used to verify the developed algorithms are presented and explained. The results for each test case are evaluated quantitatively and qualitatively. A comprehensive selection of simulation results are presented and discussed, aided by post-processing illustration of flow field variables of velocity vectors and pressure contours. The developed algorithm is successfully applied to fluid flow through wire mesh and the model has shown convincing capability in producing consistent and promising results.

6. Conclusion and Recommendations

Chapter 6 contains the conclusions for the present study. Suggestions for future work are also outlined.

References

A1 Appendix 1 : Relevant Mathematical Bookwork

A2 Appendix 2 : Publications

A3 Appendix 3 : Program Manual

A4 Appendix 4 : Program Listing

A5 Appendix 5 : Sample Input File

A6 Appendix 6 : Sample Output File

CHAPTER 2

LITERATURE REVIEW

2.1 Weave Patterns of Wire Meshes

Wire mesh is obtained by weaving wires tightly parallel between a so-called warp wire and wires lying perpendicularly known as weft wire. The wire mesh construction is defined by the kind of weave pattern. The weave pattern is determined by the sequence of intersections between the warp and weft wires. The most common and popularly used weave patterns and their characteristics are discussed in the ensuing section.

2.1.1 Characteristics of Various Weave Patterns

A plain weave is the most basic, with each warp wire passing over and under the successive weft wires and vice versa resulting in a precise mesh opening and maximum cloth stability. Square or rectangular openings are available to suit different applications and process requirements.

A twill weave produces a diagonal or twill line across the face of the fabric as the warp wire passes over and under two weft wires and vice versa. Commonly used twill weave is designated 2/1, 2/2 or 3/1 depending on how many weft wires and warp wires go over and under. This type of weave allows the use of large wire diameters to meet process requirements and is the predominant weave pattern used in filtration because of the surface area it offers.

A satin or tela weave, has the warp wires going over a number of weft wires and vice versa. Intersections between warp and weft are kept to a minimum where it is just sufficient to hold the wire firmly together. This weave permits the use of extremely strong wires with fine mesh openings resulting in a smooth and wear resistant surface. This type of weave is mainly used in vacuum filters, centrifuges, dewatering sieves and distillation columns. Plain, twill and satin weaves are the three basic weaves and are illustrated in Figure 2.1.

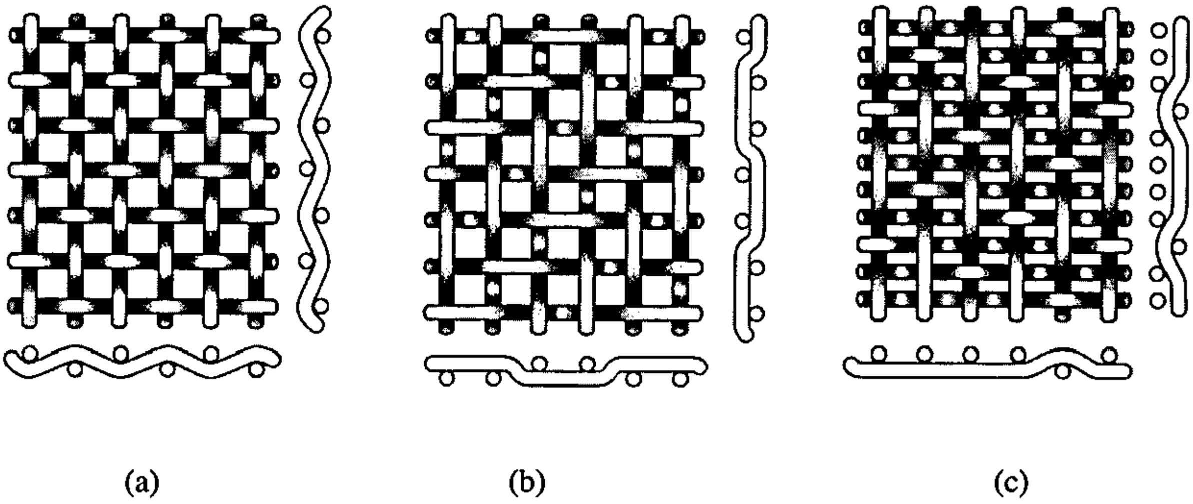


Figure 2.1 Basic configuration of three basic weaves: (a) plain weave (b) twill weave and (c) satin weave.

Apart from the basic weaves, there are other weaving patterns that are popularly used in the industry. A plain Dutch weave has a similar weave pattern to plain weave, however the warp wires are spaced farther apart than the weft wires. While the warp wires remain straight, the weft wires are plain woven to lie as close as possible against each other in a linen weave forming a dense strong material with small, irregular and twisting passageways that appear triangular when diagonally viewing the weave. It is mainly used for precoat filters and vacuum filters.

A Dutch twilled weave is a Dutch weave woven in a twill pattern that enables twice as many weft wires to be woven in the same area as a plain Dutch weave to achieve a finer filter opening. There are no apertures in the true sense of the word as the filtrate follows a sinuous path through the depth of the wire cloth. It is typically produced for very fine filtration application, ranging from 10 microns and finer.

For a reversed plain Dutch weave, the warp wires are smaller and spaced closer together than the weft wires. This weave pattern allows the use of high tensile wires and offers the most durable and stable fine filter mesh. The weave patterns of plain Dutch, Dutch twilled and reversed plain Dutch weaves are shown in Figure 2.2.

As other weaves described overleaf, a reversed Dutch twilled weave is woven under the same preferences as a reversed plain Dutch weave, but given the twilled pattern, it gives special versions of extremely rigid cloths with filtration grade down to 45 microns.

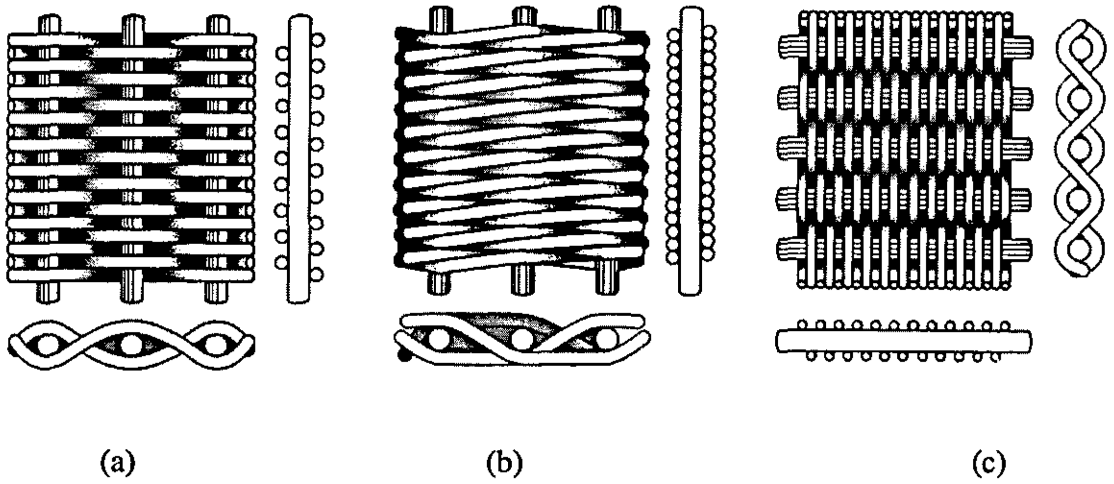


Figure 2.2 Other popular weave pattern : (a) Plain Dutch Weave (b) Dutch Twilled Weave and (c) Reversed Plain Dutch Weave.

Other types of weaving used, though not as popular as the earlier described weaving patterns, includes scapling weave (or the lock-crimp), double lock crimp, flat top, triple shoot and the intermediate crimp. The detailed description and illustration of these weave patterns can be obtained from established wire mesh manufacturers and will not be discussed in further details in this study.

There are five basic variables describing a wire mesh screen in a monofilament wire mesh (Pederson, 1969). They are:

(i) the end count (*ec*), i.e. the number of warp wires per unit length

(ii) the warp wire diameter

(iii) the pick count (*pc*), i.e. the number of weft wires per unit length

(iv) the weft wire diameter

(v) the weave pattern

These variables interact to form a pattern of either regularly or irregularly shaped holes or openings through which the fluid flows past the mesh screen, and determines the aperture size as well as the strength of the wire mesh. For any mesh, a thicker wire will provide a tougher weave, but the aperture and the open area will be reduced, giving a slower flow rate to the material passing through. These variables are therefore essentially decided depending on the applications and features required for a suitable mesh screen.

For illustration purposes, a magnified view of an industrial grade plain weave steel wire with aperture 1169 μm and diameter 0.22 mm is shown in Figure 2.3.

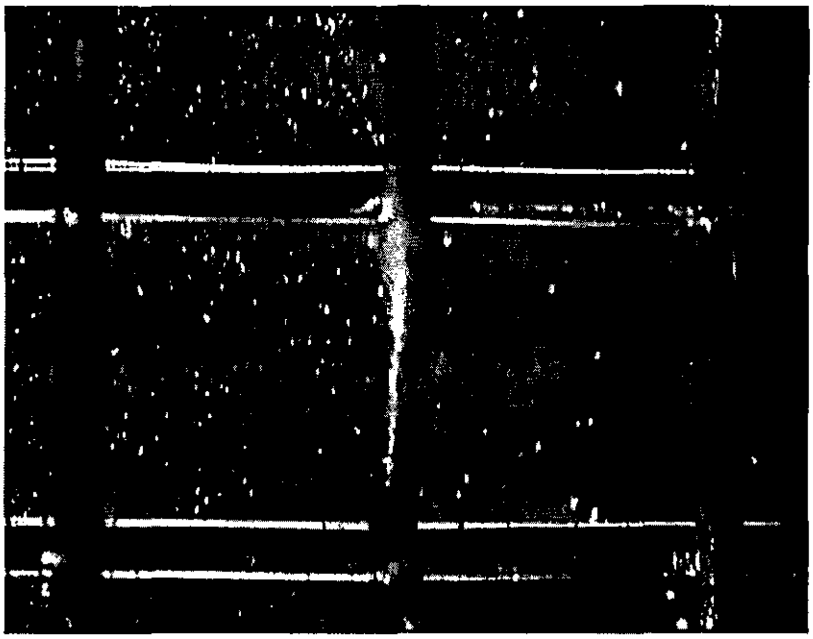
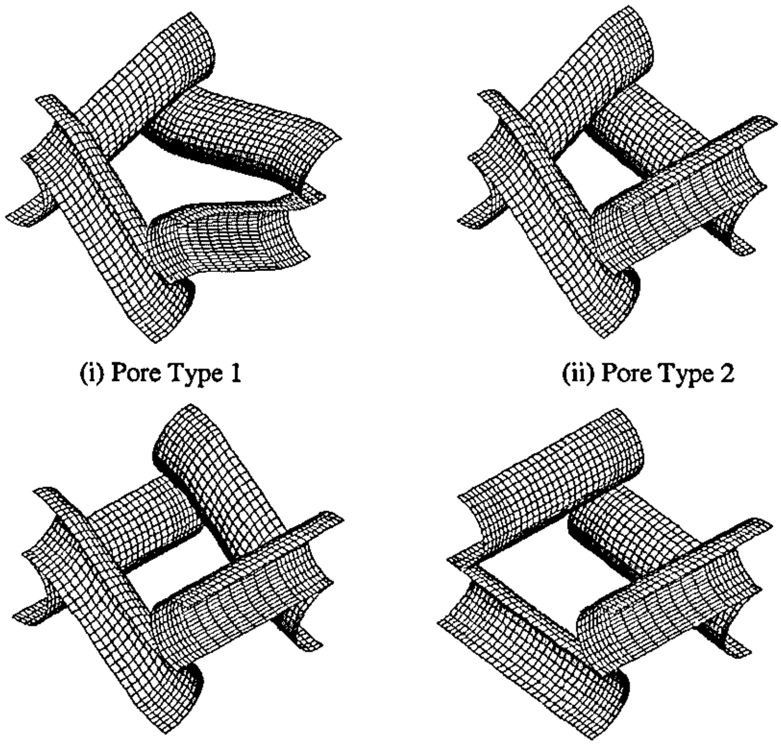


Figure 2.3 A magnified view of a plain weave type wire mesh.

2.1.2 Resulting Pore Types

Backer (1951) was the first researcher to recognize the effect of weave on pore construction and minimum cross-sectional area. He classified the pores occurring in a single layer textile fabric on the basis of yarn intermeshing. Four types of basic pore patterns, as shown in Figure 2.4, can be found in the plain weave, the twill weave, and the satin weave.



(i) Pore Type 1

(ii) Pore Type 2

(iii) Pore Type 3

(iv) Pore Type 4

Figure 2.4 Four types of basic weave patterns of a wire mesh (Backer, 1951).

The pore patterns shown in Figure 2.4 have a definable open area in the direction of the flow. For Dutch weave, Dutch twilled weave and reversed Dutch twilled weave, the pores are not as straightforward as they normally do not possess an open area in the direction of the flow but filtration is achieved through the small triangular interstices between the wires at an angle to the direction of the flow. In this study, only the four basic pore types illustrated in Figure 2.4 were considered.

2.2 Relevant Models Describing Flow Through A Wire Mesh

Theoretically, the relationship between fluid flow and the pressure difference across a fibrous or woven medium can be given by Darcy's law, which is written as (Wakeman and Tarleton, 1999)

$$u = -\frac{k}{\mu}\frac{dp}{dz} \tag{2.1}$$

where *dp* is the dynamic pressure difference across the media, *dz* the thickness of porous media, *k* the permeability, *u* the superficial velocity and μ the viscosity. However, it is widely accepted that there is still an unresolved practical problem to relate the permeability of a clean medium to its structural properties such as fibre diameter and weave construction parameters (Wakeman and Tarleton, 1999). In view of this, previous investigations have approached the problem by comparing the flow situation in the woven wire mesh to a number of analogies, a strategy common in solving fluid flow and process engineering problems. These analogies can be divided into three main categories as follows (Rushton and Griffiths, 1971):

(i) an assembly of orifices

(ii) a randomly packed bed

(iii) creeping flow over a series of cylinders

The theory and industrial significance for each of the analogies listed above will be briefly discussed in the section that follows.

2.2.1 Orifice Analogy

An orifice meter, which can be arranged by clamping a thin flat plate between flanges at a joint in a pipe as shown in Figure 2.5, is frequently used for measuring the flow of fluids. It has advantages for ordinary plant practice compared to a venturi meter as it occupies considerably less space, is inexpensive, its ratio of throat diameter to pipe diameter can be altered, and the measurable flow range is larger. The equation for an orifice meter in a steady and incompressible flow can be represented by (Bird *et al.*, 2002)

$$Q = \frac{CA_2}{\sqrt{1-(A_2/A_1)^2}}\sqrt{\frac{2(p_1-p_2)}{\rho}} \tag{2.2}$$

where Q is the volumetric flow rate, C the dimensionless discharge coefficient, A the cross section area, p the pressure and ρ the fluid density. Coefficient C is always determined experimentally and it varies considerably with changes in the ratio of orifice diameter to pipe diameter and with Reynolds numbers at the orifice (McCabe *et al.* 2001). The Reynolds number Re_o is defined as

$$Re_o = \frac{D_o u_o \rho}{\mu} \tag{2.3}$$

where D_0 is the orifice diameter and u_0 the velocity through orifice.

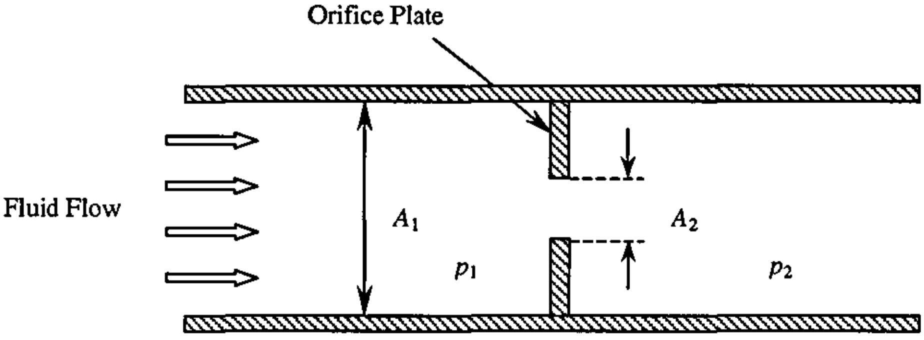


Figure 2.5 A schematic diagram of fluid flow through an orifice.

2.2.2 Random Packed Bed Analogy

Another approach to the problem has been to consider the wire mesh as a bed of cylindrical packings. Packed beds are extensively used in the chemical industries as reactors, dryers, filters, heat exchangers and adsorbers or desorbers. A considerable literature on macroscopic behaviour of flow through a packed bed exists (Mauret and Renaud 1997a, b; Dolej and Siska, 2000; Chhabra *et al.*, 2001) The flow of non-Newtonian fluids through packed bed work has been reported by Kemblowski *et al.* (1989), Chhabra (1993a, b, c) and Wu and Pruess (1996). A range of model packed beds including beds of uniform size and of multi-size spheres, non-spherical particles, mats, foams, screens, core samples and cartridges. Each porous medium is unique in its geometrical morphology, thereby contributing in certain measure to the complexity of the problem of assigning geometrical description and of making cross-comparisons between different studies. Additional difficulties arise from the significant variation in macroscopic description in terms of porosity, permeability and tortuosity of nominally similar media. Undoubtedly, the major research effort has been directed at developing simple and reliable methods of predicting the frictional pressure loss for the fluid flow through packed beds (Chhabra *et al.* 2001).

There are many approaches in predicting the frictional pressure loss in a packed bed. One of them, perhaps most popularly used, is the capillary bundle approach. In this approach, the interstitial void space in the bed of particles is envisioned to form tortuous flow passages of complex cross-sections but a constant average area for flow, as shown in Figure 2.6. Within the general framework of viscous flow regime, there are three different models that differ from each other in minor details relating to the choice of the characteristics velocity and the length of the flow passage in the main direction of flow. They are the Blake, the Blake-Kozeny and the Kozeny-Carman models. In the Kozeny-Carman model, which is widely recognized as the best model, the bed is replaced by a bundle of tubes of complex cross-section characterized by its hydraulic radius. For a Newtonian fluid, the friction factor and the Reynolds number are defined as (Chhabra *et al.*, 2001)

$$f = \frac{d_p(\Delta p)}{L(\rho u^2)} \cdot \frac{\varepsilon^3}{(1-\varepsilon)} \tag{2.4}$$

and

$$Re = \frac{\rho u d_p}{\mu(1-\varepsilon)} \tag{2.5}$$

where L is the height of bed and d_p the particle diameter.

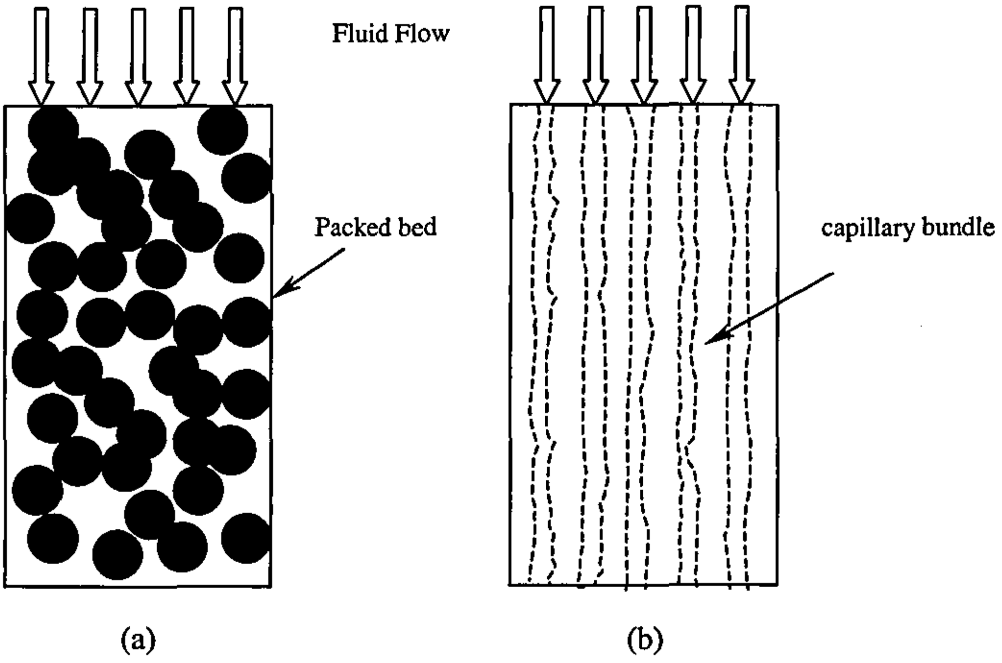


Figure 2.6 Schematic diagrams showing (a) a column packed with spheres (b) a 'capillary bundle' model for the packed column in (a).

2.2.3 Flow Over Cylinder Analogy

An alternative to the two approaches outlined in the previous sections, the flow over wire mesh can be compared to flow over an array of cylinders. Fluid flow past a periodic array of cylinders has received considerable attention due to the variety of engineering applications including flow on the shell side of tubular heat exchangers, autoclave process of manufacturing fiber-reinforced composites, enhanced oil recovery and composite and textile coating operations. A benchmark problem that has been studied extensively in the fluid mechanics community is the pressure-driven flow around a cylinder kept between parallel plates (Happel and Brenner, 1973), which was first solved analytically for Newtonian fluids by Faxén (1946). Since then, numerous researchers have investigated the flow past a single cylinder and flow over a series of cylinders (Drummond and Tahir, 1984; Hartt and Baird, 1996; Rahli *et al.*, 1996; Mauret and Renoud, 1997; Liu *et al.*, 1998; Mitsoulis, 1998; Satheesh *et al.*, 1999; Zisis and Mitsoulis, 2002), including flow of a non-Newtonian power law fluid over cylinders (Tripathi and Chhabra, 1992; Shibu *et al.*, 2001; Skartsis *et al.*, 1992; Tripathi and Chhabra, 1996; Vijaysri *et al.*, 1999; Rao and Chhabra, 2003). Studies on packed beds of cylinders have emphasized primarily the macroscopic response in the terms of pressure drop across the flow device and the general structure of the flow kinematics. The connections between the configurational changes of the macromolecules and macroscopic flow behaviour have not been a subject of primary investigation. A schematic diagram depicting the fluid flow past a submerged cylinder is shown in Figure 2.7.

The drag coefficient and Reynolds number were defined as (Wieghardt, 1953)

$$C_D = \frac{2\Delta p}{\rho u^2} \cdot \frac{\varepsilon^2}{(1-\varepsilon)} \tag{2.6}$$

and

$$Re = \frac{\rho(u/\varepsilon)d}{\mu} \tag{2.7}$$

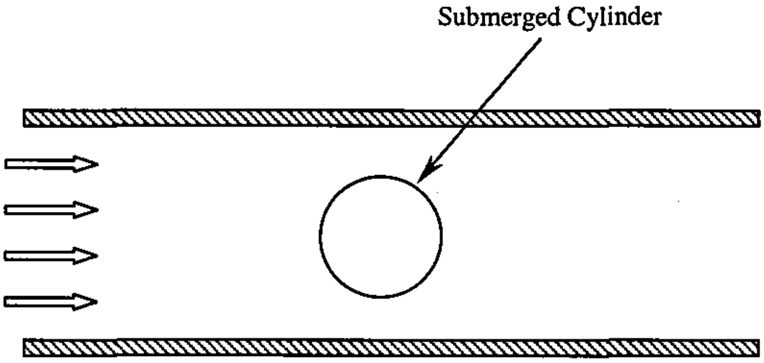


Figure 2.7: A schematic diagram showing flow past a submerged cylinder bounded by two parallel plates.

2.3 Non-Newtonian Fluid Flow Through A Wire Mesh

It has been widely recognized that rheology has a predominant role in the modelling of non-Newtonian fluid processes. Therefore, it is essential to understand the classifications of non-Newtonian fluid properties, as these will be directly relevant in this study. There are many texts describing the subject of non-Newtonian fluids properties, such as Skelland (1967), Harris (1977) and Crochet *et al.* (1984). A brief description of the non-Newtonian fluids is given in the following section.

2.3.1 Properties of Non-Newtonian Fluid

Non-Newtonian liquids are liquids that do not have a linear relationship between shear stress $\tau\,(=\mu\, dv/dy\,)$ and velocity gradient or strain rate $\dot{\gamma}\,(=dv/dy)$. There are three broad groups of non-Newtonian materials (Skelland, 1967):

(i) Time-independent fluids : sometimes referred to as non-Newtonian viscous fluid or purely viscous fluids where the rate of shear at a given point is solely dependent upon the instantaneous shear stress at that point. These materials include Bingham plastics, pseudoplastic or shear thinning fluids and dilatant or shear thickening fluids.

(ii) Time-dependent fluids : shear rate is a function of both the magnitude and the duration of shear, and also possibly the time lapse between consecutive applications of shear stress. These materials are classified into two groups: thixotropic fluids where the substances exhibit a reversible decrease in shear stress with time at a constant rate of shear and fixed temperature, and rheopectic fluids or the anti-thixotropic fluids where a reversible increase in shear stress with time at a constant rate of shear and fixed temperature can be observed.

(iii) Viscoelastic fluids : these materials show partial elastic recovery upon the removal of a deforming shear stress, and possess properties of both fluids and elastic solids. In other words, viscoelastic fluids flow when subjected to stress but part of the deformation is gradually recovered upon removal of the stress.

Examples for each category of non-Newtonian fluids are given in Table 2.1.

Table 2.1 Examples of materials for each category of non-Newtonian fluids.

Fluid Category	**Classified Group**	**Example of Materials** **(Skelland, 1967)**
Time independent	Bingham plastic	Ores, margarine, toothpaste, chocolate mixture, paper pulp.
	Pseudoplastic fluids	Adhesives, greases, mayonnaise, polymer solutions or melts, biological fluids.
	Dilatant fluids	Quicksand, wet beach sand, water-corn starch mixture.
Time-dependent	Thixotropic fluids	Oil well drilling mud, printing inks, paints.
	Rheopectic fluids	Bentonite clay suspensions, gypsum suspensions, dilute suspensions of ammonium oleate.
Viscoplastic		Bitumen, flour dough, Napalm, jellies.

There are many models such as the Power Law Model, the Carreau Model and the Maxwell Model used in the representation of apparent viscosity η in non-Newtonian liquids. However, the Power Law model gives a representation of η for a large number of systems over a wide range of shear rates. Moreover, its relatively simple form facilitates the necessary computation work. Further discussions on the constitutive equations will be covered in Chapter 3. The flow curves typical of Newtonian fluid and time-independent fluids are sketched in Figure 2.8.

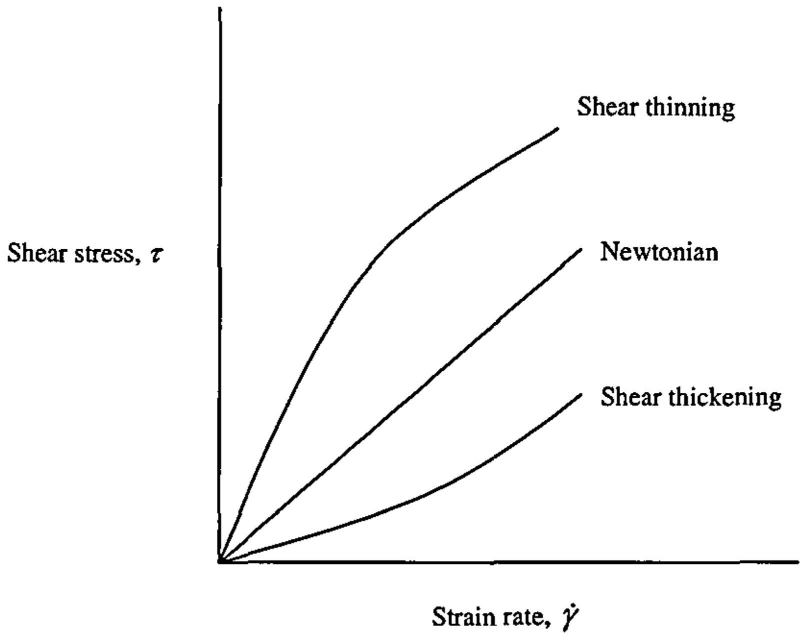


Figure 2.8 Flow curves of Newtonian and time-independent fluids.

The majority of polymers and non-Newtonian materials encountered in industry fall in the category of pseudoplastic fluids or shear thinning fluids, while dilatant fluids are less common. In this study, the hydrodynamic properties of a fire retardant fluid which is shear thickening and is used in aeronautical applications will be used as the primary working fluid. Chosen primarily for its non-hydrocarbon based properties for aircraft safety reasons, this fluid copes with extreme temperature conditions and rapid temperature change without compromising its performance. The power law index *n* of this fluid will be manipulated to reflect the rheological behaviour of Newtonian, shear thinning and shear thickening effect as the fluid flow past a wire mesh.

2.3.2 Experimental Studies

Over the years, many studies have been conducted to investigate fluid flow past ideal shapes such as cylinders and spheres for Newtonian fluids or non-Newtonian fluids (Skelland, 1967; Astarita & Marrucci, 1974; Schowalter, 1978; Chhabra, 1999; Chhabra & Richardson, 1999). For airflow through fabrics, studies to correlate air permeability to a range of materials, the effects of fabric tension on permeability, the variations of permeability with pressure induced tension, and the effects of material weight on airflow resistance have also been conducted for applications such as textile, filter media and parachute design (Rainard, 1946; Rainard, 1947; Cunningham *et al.*, 1954; Goglia *et al.* 1955; Baker, 1956; Davies, 1973; Skelton and Abbott, 1974; Payne, 1978). However, studies on woven wire screens have been limited and they typically considered permeability, porosity and resistance properties of the wire mesh. Fluid flow through screens have been studied by various researchers such as Wieghardt (1953), Armour and Connon (1968), Rushton (1969), Rushton and Griffiths (1971), Ehrhardt (1983), Squiers (1984), Chhabra and Richardson (1985), Kiljański and Dziubiński (1996). The majority of researchers used various gases and Newtonian liquids as the experimental media in their work.

Backer (1948) suggested that the pores between the woven wires form a series of orifices or nozzles through which the flow motion took place. A woven wire mesh is simply a flow barrier formed by a large number of orifices or nozzles. Robertson (1950) adopted the idea and plotted experimental values of log C against log Re for airflow through plain weave metallic meshes. He obtained a good correlation between these two variables and went on to apply the idea to analyze loosely woven multifilament cloths of differing weave pattern. The correlations he obtained in this case, however, were not as good as those he had earlier obtained for the plain weave monofilament meshes. This brought Backer (1951) to realize that the single most important factor influencing the flow rate when fluid flows through an orifice was the minimum cross section area of the pore. Undeterred by other researcher's criticisms that the interstices of the pores are too complicated for a complete hydrodynamic analysis, Backer set out a classification of the basic pore types in a woven fabric on the basis of the mode of yarn intermesh (shown earlier in Figure 2.4). This was his first step towards a detailed and significant analysis of the cross-sectional area. To deal with the pore areas in a more quantitative manner, he considered the geometry of the unit cell to comprise of one of two geometric forms, which are the circular cylinder and the torus. Although the analytical forms for the cross-sectional areas of the two separate forms are well developed, the expression for the horizontal cross-sectional area for both the cylinder and the torus are complicated. To overcome this, Backer adopted a mathematical integration using a modelling technique. Slices of each basic pore type were taken at successive depths and the cross-sectional areas revealed were determined by mechanical integration. The unit pore area was obtained by deducting the cross-sectional area of the yarn from the cross-sectional area of the unit cell. Using this technique, the minimum cross-sectional area of each basic pore type was successfully determined. Backer found the use of minimum pore area greatly reduced the scatter produced by calculations based on the projected open area.

Prior to the work of Wieghardt (1953), researchers have consistently used the approach velocity u_∞ in computing the drag or resistance coefficient for flow through wire mesh. This often resulted in various curves for various values of porosity ε. Wieghardt (1953) correctly pointed out in his paper that the characteristic velocity for the flow should be the hydraulic velocity, given by u/ε, which is the higher velocity when the fluid flows through the mesh. By adopting the flow around single cylinder analogy, he proposed a Reynolds number *Re* and a dimensionless drag coefficient C_D given earlier in equations (2.6) and (2.7). Using these equations, he recalculated data for airflow through metal wire mesh from previous researchers and found that the majority of the data fitted well in a common correlation, given as $C_D = 6.5/Re^{1/3}$ for $60 \leq Re \leq 600$, with the corresponding porosity ε in the range $0.318 \leq \varepsilon \leq 0.682$. The correlation does not seem to interpret the data very well beyond the given range. He went on to suggest that the orifice analogy could be equally adopted in analyzing the flow through a wire mesh, but insisting that the flow around an infinite cylinder would be more fitting as the surface for the wire mesh is rounded. He however failed to recognize the influence of the weave pattern on the pressure drop thus fell short of reporting the pattern of wire mesh weave in his study.

Armour and Cannon (1968) adopted a slightly different approach in their experimental study, in which the flow of nitrogen and helium gas through a woven metal mesh was compared to flow around a thin packed bed of spheres. They investigated the effect of different weave patterns including plain, twill, plain Dutch and Dutch twill weaves on pressure drop. Applauding the earlier investigation by Ingmanson (1961) who appeared to be the first researcher to recognize the three-dimensional property of wire screens by characterizing the screen geometry using the surface area to unit volume ratio and the void volume, they derived a series of equations to calculate the surface area to unit volume ratio and void fraction for various weave patterns mentioned earlier. They went on to incorporate these variables into the equations determining the friction factor f and the Reynolds number. Using these equations with which to analyze their experimental data, they managed to find a correlation of $f = (8.61/Re) + 0.52$ for the range $1 \leq Re \leq 100$. They also claimed that the correlation works well when applied to previous researchers' data for flow of water through a semi-twill weave in the range $1 \leq Re \leq 1000$. Interestingly, their data showed a consistently lower pressure drop for a plain weave compared to a twill weave, and a consistently lower pressure drop for a Dutch twill weave compared to a plain Dutch weave. Their contribution was perhaps useful in developing a predictive tool for the design engineer to predict the pressure drop across wire meshes for different weave patterns using the specifications provided by the manufacturers.

Pederson (1969) further clarified the orifice analogy by stating that each irregularly-shaped opening in a simple monofilament weave can be considered as an orifice. He advocated the idea of utilizing the 'effective area' and the 'effective diameter' to describe the fluid flow through an orifice. The effective diameter is defined as

$$D' = 4\frac{A}{W} \tag{2.8}$$

where A is the effective area of an orifice and W the wetted orifice perimeter where the flow is most constricted. He then devised a method to reduce the five basic variables describing a woven fabric, discussed in the earlier section, to two significant variables A and W. He used a discharge coefficient given by

$$C = \left[\frac{\rho V^2}{2\Delta p} \cdot \frac{\left(1-\alpha^2\right)}{\alpha^2}\right]^{1/2} \tag{2.9}$$

where the effective fraction open area is defined as

$$\alpha = A(ec)(pc) \tag{2.10}$$

where *ec* is the end count and *pc* is the pick count. The discharge coefficient was given as

$$C = f'\left[\frac{4\rho V}{W(ec)(pc)\mu}\right] \tag{2.11}$$

He went on to determine *A* and *W* analytically for both plain weave and twill weave pore. He then plotted log *C* versus log *Re* for airflow through a wide range of plain weave and twill weave monofilament fabrics and found the correlation to be excellent.

Rushton and Griffiths (1971) conducted a series of experiments to investigate the effect of pore structure on pressure drop with water flow through monofilament cloths. They used plain weave, twill weave, and satin weave cloths made of Nylon and Polypropylene in their study, including a double warp twill weave and a double warp plain weave. They comprehensively presented various approaches to explaining the flow situation in a woven fabric including the orifice analogy, the randomly packed bed analogy and the flow over cylinders analogy. Realizing that some twill and satin weaves consist of two or three different basic pore patterns side by side in a single fabric, they adopted the idea of a 'unit cell', which defines the ratio of each pore type everywhere in the fabric. The ratio of this 'unit cell' is then used to determine the average effective fraction open area of the fabric. They plotted experimentally determined values of log *C* against log *Re* for water flow through woven cloths and observed that $C = 0.17\ Re^{0.41}$ can be used to predict the pressure drop in the range $1 < Re < 10$. They also observed that the plain weave cloths generally show a better degree of correlation than the twill as the plain weave were nearly perfectly woven but the twill weaves were irregular. It can be observed from their plots that the plain weave generally gives the lowest pressure drop while the twill weave gives the highest pressure drop, with the satin weave ranging between these two weaves. This is consistent with the observation in the results presented by Armour and Cannon (1968). Based on their analysis, they claimed that Pederson's analysis (1969), which is based on the orifice analogy, is more useful where it successfully predicted resistance in more complex cloths of widely differing yarn configuration whereas the randomly packed bed analogy was only useful for plain weave prediction as large errors occurs when it is applied for twill weave cloths.

Motivated by Wieghardt's (1953) work, Ehrhardt (1983) extended the study to a wider range of wire meshes and fluids. He used 61 different square mesh weaves and 95 different Dutch weaves in spindle oil, diesel oil, petroleum, water and air in laminar and turbulent flow regimes. He utilized two different types of apparatus to carry out the study for air flow and liquid flow, respectively. Using the same approach and equations proposed by Wieghardt, he found his experimental data fell in good agreement with Wieghardt's earlier analysis. He picked up where Wieghardt had failed in proposing a correlation that includes the transitional flow and turbulent flow regime. The correlation $C_D = 0.72 + 49/Re$ for $0 \le Re \le 1000$, with the porosity in the range $0.25 \le \varepsilon \le 0.682$ was found to give good predictions for pressure drop across wire meshes. Although he failed to report the wire mesh weave pattern in his study, it can be observed from his analysis that ε has a significant effect on the C_D where higher ε rendered lower C_D. He also found that upstream turbulence can reduce C_D by 20% if $Re > 100$ while bubbles that adhere to the cloth can increase C_D by up to 40%.

Investigations of non-Newtonian fluid flow through various single screens have been extremely limited and mostly concentrated on shear thinning fluids. A detailed literature search has not found any reported study conducted on shear thickening fluids. Chhabra and Richardson (1985) were the first researchers to study non-Newtonian fluid flow past a single screen. They used aqueous solutions of glucose for a Newtonian fluid and carboxymethyl cellulose (CMC) solutions of various concentrations with n between 0.34 to 0.61 as shear thinning fluids in their experiments, as given in Table 2.1. Using the analogy of flow around a cylinder approach, they adopted the empirical equations proposed by Wieghardt (1953) to interpret their experimental results for wire meshes with porosities $0.34 \leq \varepsilon \leq 0.36$. They found the results for Newtonian fluid were in good agreement with the correlation for ψ proposed by Ehrhardt (1983) for the range $10^{-5} \leq Re \leq 10^{3}$. For the non-Newtonian fluid, they used a modified Reynolds number for non-Newtonian fluid in which Re_{NN} was expressed as

$$Re_{NN} = \frac{\rho (u/\varepsilon)^{2-n} d^{n}}{K} \tag{2.12}$$

Using this modified Re_{NN} while maintaining the equation for C_D as that of the Newtonian fluid, they found the results for shear thinning fluids can be expressed by $C_D = 50/Re_{NN}$, which is very close to the correlation $C_D = 0.72 + 49/Re$ (Ehrhardt, 1983) where the first term can be justifiably ignored for creeping flow when $Re < 1$. The range of Re_{NN} in their study for non-Newtonian fluid was $5*10^{-7} \leq Re \leq 10^{-3}$, as shown in Figure 2.1. Encouraged by their results, they extended the experiments to study shear thinning suspensions of kaolin in water. However, the rapid build up of filter cake reduced the flow rate significantly and the problem rendered the apparatus unsuitable for study of suspensions.

Table 2.2 Physical properties of the test fluids and characteristic dimensions of the plain weave wire mesh screens and monofilament cloth (Chhabra and Richardson, 1985).

Test Liquid	n	η_0 (Pa s)	Aperture (μm)	d (μm)	Symbols in Figure 2.9
1.5% CMC in water	0.60	6.6	53	36	■
1.5% CMC in water	0.59	7.6	53	36	▲
2.0% CMC in water	0.61	9.5	150	100	△
2.5% CMC in water	0.34	45.0	150	100	○
2.5% CMC in water	0.34	76.0	355	280	●

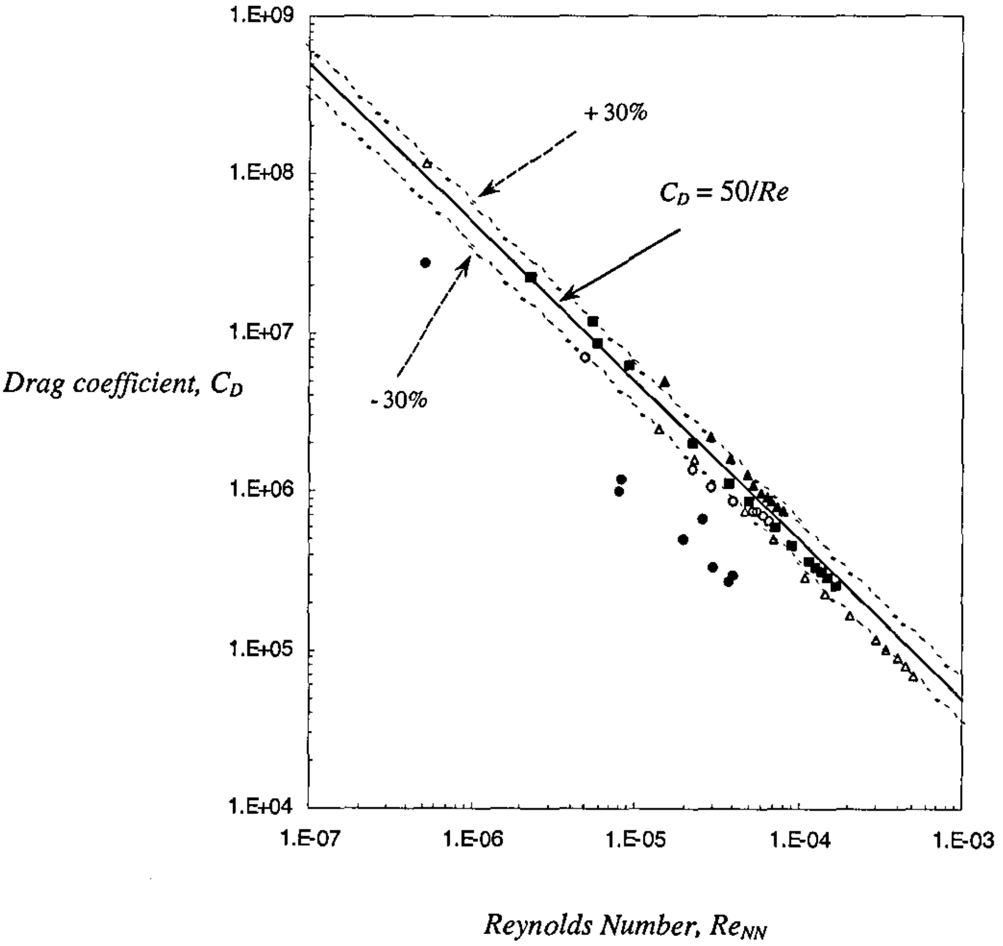


Figure 2.9 Drag coefficient versus Reynolds number for non-Newtonian fluids (Chhabra and Richardson, 1985).

Kiljański and Dziubiński (1996) furthered the study to include sets of wire meshes using polyethylene terephthalate (PTEF) and potato syrup for a Newtonian fluid and low-density polyethylene (LDPE) for a shear thinning fluid. They used a range of meshes with $0.256 \leq \varepsilon \leq 0.414$ to test for a lower range of *Re*. They adopted Wieghardt's empirical equations and found the dependence of C_D on the *Re* for Newtonian fluid flow through a single screen can be described by $C_D = 62/Re$ for the range $10^{-8} \leq Re \leq 10^{-5}$. For a pack of wire screens, which consists of either identical or different wire meshes being packed together, the correlation was found to be $C_D = 53/Re$. They further found that the resistance to flow through one screen in a set of packing is about 17% lower than the resistance to flow through a single wire mesh. Also, if the number of wire meshes in the packing exceeds ten, a further increase in the number of wire meshes will result in a proportional increase in pressure drop. For non-Newtonian fluid, the correlation was found to be $C_D = 41/Re_{NN}$ for the range $10^{-11} \leq Re \leq 10^{-6}$.

A summary of the correlations proposed by various researchers in the corresponding *Re* and ε has been given in Table 2.3. None of the researchers showed an attempt to capture the velocity profile when the fluids flow past the wire mesh interstices.

Table 2.3 A comparison of approaches and correlations proposed by previous researches.

Researchers	**Screen Material**	**Weave Pattern**	**Fluids**	**Approach**	**ε range**	**Re or Re_{NN} range**	**Proposed Correlation**
Wieghardt (1953)	Metal screen	Not specified	Newtonian (air)	Flow over cylinder	$0.318 \leq \varepsilon \leq 0.682$	$60 \leq Re \leq 600$	$C_D = 6.5/Re^{1/3}$
Armour and Cannon (1968)	Metal screen	Plain, Twill, Plain Dutch, Twill Dutch	Newtonian (Nitrogen and helium)	Packed bed	$0.165 \leq \varepsilon \leq 0.430$ (excluding Dutch weave)	$1 \leq Re \leq 100$	$f = 0.52 + 8.61/Re$
Rushton and Griffiths (1971)	Nylon and Polypropylene cloths	Plain, Twill, Satin	Newtonian (Water)	Orifice	$0.277 \leq \varepsilon \leq 0.487$	$1 < Re < 10$	$C_D = 0.17Re^{0.41}$
Ehrhardt (1983)	Metal screen	Square mesh and Dutch weave	Newtonian (Spindle oil, diesel oil, petroleum, water, air)	Flow over cylinder	$0.250 \leq \varepsilon \leq 0.682$	$0 \leq Re \leq 1000$	$C_D = 0.72+49/Re$
Chhabra and Richardson (1985)	Metal screen	Not specified	Non-Newtonian (CMC)	Flow over cylinder	$0.340 \leq \varepsilon \leq 0.360$	$5*10^{-7} \leq Re \leq 10^{-3}$	$C_D = 50/Re_{NN}$
Kiljański and Dziubiński (1996)	Metal screen and packing of screens	Not specified	Non-Newtonian (LDPE)	Flow over cylinder	$0.256 \leq \varepsilon \leq 0.414$	$10^{-8} \leq Re \leq 10^{-5}$	$C_D = 41/Re_{NN}$ *(packing of screens)*

2.3.3 Numerical Studies

Although CFD has been widely used in studying many aspects of the fluid flow processes, there has been very limited study of fluid flow through wire meshes using numerical methods. No significant attempt has been made to predict the flow field within the mesh interstices although such studies have been widely applied in other fields. Among the few researchers who studied fluid flow through screen numerically are Lu *et al.* (1996) and Tung *et al.* (2002), who used water in their respective studies.

Lu *et al.* (1996) lamented that most previous studies on fluid flow through woven structures have focused on the pressure drop problem and not the macroscopic details of velocity profile and pressure contour in the interstices. They argued that a better understanding of the flow pattern in the woven structure could prove useful in examining the initial stages of cake filtration as well as the effect of weaves on fouling phenomena within a filter cloth. They studied the effect of fabric pore construction, aperture and Reynolds number on the flow pattern in the interstices and downstream of a cloth using the commercial CFD software FLUENT^{TM}. They discovered that the flow pattern in the interstices and downstream were different for each basic pore type. As the position of the upper filament in each pore type differs, the flow of water which swirls around the upper surfaces of the upper filament into the narrow channel between the cloth filaments was characteristically unique for each case. They adopted the orifice approach used by Pederson (1969) and Rushton and Griffiths (1971) to analyze the fluid resistance results. Their simulations showed that as the aperture decreases, there is a corresponding decline in the filtrate flux under a constant operating pressure. Using the equations proposed by Pederson

(1969), they plotted log C_D versus log *Re* graph and obtained a correlation to describe their data, given as

$$C_D = 0.18\text{Re}^{0.49} \tag{2.13}$$

for the range $6*10^{-2} \leq Re \leq 20$. Their predictions also showed that among the basic pore models with the same yarn diameter and spacing, pore type 1 gives the highest resistance, followed by pore type 2, 3 and 4. By weighting the contribution of pore types in each weave pattern, they concluded that a fabric with plain weave is most resistant to the fluid flow, while satin weave shows the least resistance. However, they failed to examine their simulation results against the experimental results of previous researchers to give credibility to their conclusions, which contradict the experimental results of Armour and Cannon (1968) and Rushton and Griffiths (1971).

Tung *et al.* (2002) furthered Lu *et al.*'s study to include fluid flow through multifilament cloths and spun staple yarn, which is manufactured from short fibers using spinning techniques. By assigning artificial permeability values in the range $5.0*10^{-10} \leq Re \leq 10^{-12}$ to the woven filaments, they investigated the distribution of water flow in interyarn and interfibre pores of four basic pore types. Their results indicated that a decrease in permeability of fabric yarn rendered a decrease in the filtrate flux under constant operating pressure, whereas an increase in flow rate resulted in an increase of the pressure drop. Their simulations results showed a similar pattern of pressure drop to that of Lu *et al.* where pore type 1 were found to give the highest pressure drop and pore type 4 gives the lowest pressure drop.

2.4 Closure

In this chapter, the experimental and numerical studies conducted by various researchers on fluid flow through a wire mesh has been reviewed. Investigators were found to have approached the problem by comparing the flow situation in the woven wire mesh to (i) an assembly of orifices, (ii) a randomly packed bed, and (iii) creeping flow over a series of cylinders. The relevant results and proposed correlations of the investigations were also discussed and summarized. In the next chapter, the governing equations and boundary conditions used in the mathematical modelling of this study will be discussed to set the foundation for deriving the final working equations.

CHAPTER 3

GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

3.1 Flow Model

Numerical modelling of a flow system is based on the fundamental governing equations that reflect the physics of fluid dynamics. A numerical process starts with the formulation of a mathematical model based on the law of conservation of mass, energy and momentum, as well as a rheology equation that describes the constitutive behaviour of the fluid. The general equations of non-Newtonian fluid mechanics are derived on the basis of physical law and rheological experiments. Due to the predominant role of non-Newtonian flow equations in modelling, it is therefore important to understand the theoretical foundations of these equations. In this chapter, a three-dimensional mathematical model based on flow and constitutive equations is considered in a fixed (Eulerian) coordinate system. For a laminar, isothermal flow, the flow regime is described in Cartesian coordinate systems by the following set of equations.

3.1.1. Equation of Continuity

The continuity equation for an incompressible fluid is written as

$$\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \tag{3.1}$$

where v_x, v_y *and* v_z are the components of the velocity fields. The lack of a pressure term in Equation (3.1) is a known source of numerical challenge. The interpolation functions that one can use in the numerical solution will be very much limited by the stability conditions, known as the mixed patch test or the Ladyzhenskaya-Babuŝka-

Brezzi stability condition (Ladyzhenskaya, 1969; Babuŝka, 1971; Brezzi, 1974). For this reason, much interest has been focused on the development of so called stabilized procedures in which the violation of the test is artificially compensated. Zienkiewicz and Taylor (2000) outlined several such procedures that include the introduction of non-zero diagonal terms by adding a least-square form to the Galerkin formulation (Courant, 1943; Brezzi and Pitkäranta, 1984), the introduction of so-called finite element calculus to the formulation to gain addition of diagonal terms (Oñate, 1998) and the consideration of slightly compressible form of continuity equation in order to introduce the pressure term (Zienkiewicz and Wu, 1991).

The approach suggested by Zienkiewicz and Wu has been adapted in this study. A slightly perturbed form of Equation (3.1) is considered in this study whereby the term $(1/\rho c^2)(\partial p/\partial t)$ is included to the left hand side of the equation in order to satisfy the Ladyzhenskaya-Babuŝka-Brezzi (LBB) stability condition. The continuity equation is now written as:

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t}+\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \qquad (3.2)$$

where c is the speed of sound in the fluid, p is the pressure, ρ is the fluid density and t is the time variable. The use of this slightly perturbed form of continuity equation, which corresponds to slightly compressible fluids, allows the utilization of equal order interpolation model for the velocity and pressure (Zienkiewicz and Wu, 1991) hence increases the flexibility of the developed solution scheme (Nassehi, 2002). This approach has been applied successfully to modelling crossflow membrane filtration (Nassehi, 1998), flow past porous boundaries (Richardson and Nassehi, 2003), and flow through pleated cartridge filters (Nassehi *et al.*, 2005). It should be noted that the additional term included in Equation (3.2) is relatively small hence will not adversely affect the simulation results. Furthermore, at the steady limit this term is not involved, consequently the solution will correspond to the incompressible case.

3.1 2. Momentum Equations

The equation of motion is based on Newton's second law of motion, or the law of conservation of momentum and is written as

$$\rho\frac{\partial v}{\partial t}+\rho v\cdot\nabla v=\nabla\cdot\sigma+\rho g \qquad (3.3)$$

where ∇ is the operator nabla, σ is the Cauchy stress tensor and g is the body force. The Cauchy stress tensor is given as

$$\sigma_{ij}=-p\delta_{ij}+\tau_{ij} \qquad (3.4)$$

where δ_{ij} is Kronecker delta and τ_{ij} is the extra stress tensor. In an expanded form, the normal stresses are given as

$$\sigma_{xx}=-p+\tau_{xx}\,;\ \sigma_{yy}=-p+\tau_{yy}\,;\ \sigma_{zz}=-p+\tau_{zz} \qquad (3.5a, b, c)$$

where

$$\tau_{xx}=2\eta\frac{\partial v_x}{\partial x}\,;\ \tau_{yy}=2\eta\frac{\partial v_y}{\partial y}\,;\ \tau_{zz}=2\eta\frac{\partial v_z}{\partial z} \qquad (3.6a, b, c)$$

and shear stresses given as

$$\tau_{xy}=\tau_{yx}=\eta\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right) \qquad (3.7a)$$

$$\tau_{yz}=\tau_{zy}=\eta\left(\frac{\partial v_y}{\partial z}+\frac{\partial v_z}{\partial y}\right) \qquad (3.7b)$$

$$\tau_{zx} = \tau_{xz} = \eta\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right) \tag{3.7c}$$

where η is the fluid viscosity.

The majority of highly viscous non-Newtonian fluid flows are characterized as low Reynolds number Stokes flow regimes or creeping flow where the fluid flow is very slow, normally with $Re<1$. In this flow regime, the inertia term $v.\nabla v$ in the equation of motion are neglected (Bird *et al.*, 2002; Nassehi, 2002). In addition, highly viscous flow systems are in general dominated by stress and pressure variations and in comparison the body forces acting upon them are small and can be justifiably ignored (Nassehi, 2002). Incorporating Equations (3.5), (3.6) and (3.7), the momentum conservation equation for creeping flow can be written as

$$\rho\frac{\partial v_x}{\partial t} = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \tag{3.8a}$$

$$\rho\frac{\partial v_y}{\partial t} = -\frac{\partial p}{\partial y} + \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} \tag{3.8b}$$

$$\rho\frac{\partial v_z}{\partial t} = -\frac{\partial p}{\partial z} + \frac{\partial \tau_{zx}}{\partial x} + \frac{\partial \tau_{zy}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \tag{3.8c}$$

or in the expanded form provided that Equation (3.4) can be used, i.e. the fluid is generalized Newtonian,

$$\rho\frac{\partial v_x}{\partial t} = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial x}\left[2\eta\frac{\partial v_x}{\partial x}\right] + \frac{\partial}{\partial y}\left[\eta\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\right] + \frac{\partial}{\partial z}\left[\eta\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right)\right] \tag{3.9a}$$

$$\rho\frac{\partial v_y}{\partial t} = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial x}\left[\eta\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\right] + \frac{\partial}{\partial y}\left[2\eta\frac{\partial v_y}{\partial y}\right] + \frac{\partial}{\partial z}\left[\eta\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right)\right] \tag{3.9b}$$

$$\rho\frac{\partial v_z}{\partial t} = -\frac{\partial p}{\partial z} + \frac{\partial}{\partial x}\left[\eta\left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z}\right)\right] + \frac{\partial}{\partial y}\left[\eta\left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y}\right)\right] + \frac{\partial}{\partial z}\left[2\eta\frac{\partial v_z}{\partial z}\right] \tag{3.9c}$$

This can be summarized into a simpler form,

$$\rho\frac{\partial v_i}{\partial t}=-\frac{\partial p}{\partial i}+\frac{\partial}{\partial i}\left[2\eta\frac{\partial v_i}{\partial i}\right]+\sum\frac{\partial}{\partial i}\left[\eta\left(\frac{\partial v_i}{\partial j}+\frac{\partial v_j}{\partial i}\right)\right] \tag{3.10}$$

where $i = x, y, z$ and $j \neq i$.

3.2. Constitutive Equation

The constitutive equation shows the relationship between the extra stress and the rate of deformation of the fluid. In non-Newtonian fluid mechanics, the choice of constitutive equation or rheological model depends critically on the type of flow considered and it is this basic consideration which makes non-Newtonian fluid mechanics different from classical fluid mechanics, where the Navier-Stokes equations can be immediately accepted as being valid for all flow situations (Astarita, 1976; Crochet *et al.*, 1984). Many rheological formulae have been published with the intention of describing one or more features of non-Newtonian fluid behaviour and they can be found in non-Newtonian literature such as Metzner (1961), Skelland (1967) and Harris (1977). Some of these rheological equations have limited use and therefore the application of any formulae should be done with great care, especially when the fluid motion is anything but simple steady shearing motion. The derivation of universally applicable constitutive models for non-Newtonian fluid is generally not attempted, if at all possible, as it is extremely difficult to establish quantitative relationships between the microscopic structure of non-Newtonian fluids and their macroscopic properties (Nassehi, 2002).

The expression most widely used in the less rigorous technological publications for non-Newtonian fluids is perhaps the Ostwalde-de Waele model, which is more widely known as the power law model and is written as

$$\eta = \eta_0 (\dot{\gamma})^{n-1} \tag{3.11}$$

where η_0 is the consistency index, n is the power law index and $\dot{\gamma}$ is the shear rate. For $n<1$ it is used to describe pseudoplastic behaviour. For $n>1$ it is used to describe dilatant behaviour while for $n = 1$ it reverts to the Newtonian expression. The shear-dependent viscosity of the non-Newtonian fluid in the present study, η, is calculated and updated using the power law model.

3.3. Assumptions

The essential assumptions adapted in this study to solve the described problem are discussed in the section that follows.

3.3.1 Assumptions Relating to Wire Mesh Geometry

In order that a rigorous assessment of effects of the geometric configuration of any area of wire mesh can be made, it is necessary to make the following assumptions (Pederson, 1969):

(i) the filling wires, or weft wires, are straight

(ii) the weaving is perfect

(iii) the yarns are all cylindrical

(iv) the warp wires are perfectly straight between filling wires

(v) flow through any opening is not influenced by flow through any other opening

(vi) the flow is perfectly isothermal and the woven metal wire does not expand during the flow

(vii) the integrity of the wire mesh is upheld throughout the period of simulation

These assumptions are vital in the mesh generation of the three-dimensional problem domains. The solution domains have been designed based on these assumptions in order that a realistic comparison on the different pore types can be made. The analysis of the simulation results in chapter five was also based on these essential assumptions.

3.3.2 Wall Slip

The imposition of no-slip velocity conditions at solid walls is based on the assumption that the shear stress at these surfaces always maintained below a critical value to allow a complete wetting of the wall by the fluid (Nassehi, 2002). This implies the fluid is constantly sticking to the wall and is moving with a velocity exactly equal to the wall velocity. The wall-slip phenomenon is described by Navier's slip condition, which is a relationship between the tangential component of the momentum flux at the wall and the local slip velocity (Silliman and Scriven, 1980). Using a two-dimensional domain as an illustration, this relationship is expressed as

$$[\beta\tau \cdot \hat{n} + (v - v_b)] \cdot \hat{t}^T = 0 \tag{3.12}$$

where $\hat{t}$ and $\hat{n}$ are unit vectors tangent and normal to the boundary, τ is the extra stress tensor, β is a slip coefficient, v is the fluid velocity vector and v_b is the velocity of the solid wall. Equation (3.12) together with the following equation which represents no flow through a solid wall, are used to impose slip-wall boundary conditions.

$$v \cdot \hat{n} = 0 \tag{3.13}$$

Consider a solid wall section as shown in Figure 3.1, the following relationships between the components of unit outward normal and tangential vectors are true at all points

$$\hat{t}_y = +\hat{n}_x, \quad \hat{t}_x = +\hat{n}_y \tag{3.14}$$

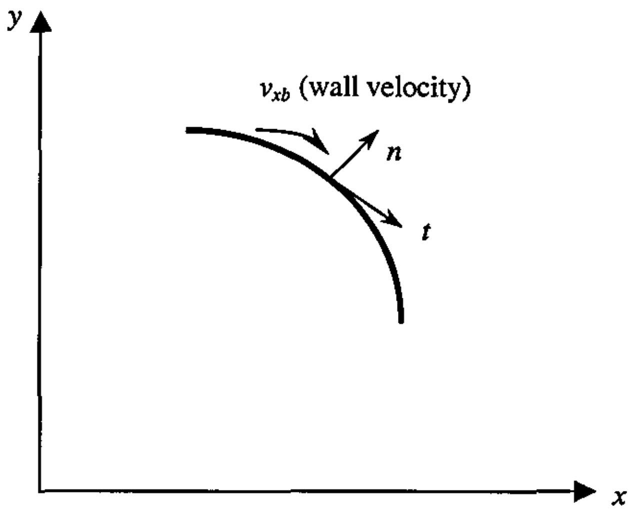


Figure 3.1 Slip at a solid wall.

Equations (3.12) and (3.13) are recast in their components and solved together. After algebraic manipulations and making use of the relationship given Equation (3.14), slip wall velocity components are given as

$$v_x - v_{xb} = -\eta\beta\hat{n}_y\left[2\left(\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y}\right)\hat{n}_x\hat{n}_y + \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\left(\hat{n}_y^{\,2} - \hat{n}_x^{\,2}\right)\right] \tag{3.15a}$$

$$v_y - v_{y_b} = -\eta\beta\hat{n}_x\left[2\left(\frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y}\right)\hat{n}_x\hat{n}_y + \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right)\left(\hat{n}_y^{\,2} - \hat{n}_x^{\,2}\right)\right] \tag{3.15b}$$

The slip coefficient β is defined as (Nassehi, 2002)

$$\beta = \frac{\beta_0}{l} \tag{3.16}$$

where β_0 is the initial slip coefficient and l is a characteristic flow domain dimension. The limit of $\beta \rightarrow 0$ corresponds to no slip ($v_x = v_{xb}$, $v_y = v_{yb}$) and the limit of $\beta \rightarrow \infty$ gives the perfect slip condition. The slip coefficient β depends on the invariants of the stress tensor and the surface roughness. Navier's slip condition can be discretized in a similar manner to the main flow equations and directly incorporated into the finite element working equations (Ghoreishy and Nassehi, 1997). Generally, on no-slip walls, zero velocity components ($v_x = v_y = v_z = 0$) can be imposed as boundary conditions.

3.4 Boundary Conditions

In order to solve the flow equations consisting of the continuity equations, momentum equations and rheological model, appropriate boundary conditions need to be prescribed. These conditions may include velocity, stress or surface force components and a datum for pressure. For Newtonian fluid mechanics, it is normally sufficient to specify the velocity or surface force components over the boundary of the domain of interest and the pressure at one point when no normal surface force has been specified anywhere on the boundary. For viscoelastic fluids, this specification is obviously insufficient as the flow is influenced by the fluid memory. This is especially important in cases where the boundary of the domain contains an inlet where the strain history of the fluid entering the domain need to be known, or equivalently, the knowledge of the stress field on entry to the domain (Crochet *et al.*, 1984; Nassehi, 2002). In practice, the boundary condition requirements are often inadvertently satisfied by assuming fully developed flow conditions at inlet that essentially implies knowledge of the flow field upstream of the domain of interest. Nassehi (2002) further suggested that in engineering simulations of non-Newtonian fluid systems, a set of conditions that can be shown to make the best possible physical sense under the given condition is usually used as complete and accurate mathematical evaluations of such boundary conditions are in general not possible.

There are basically two types of conditions that need to be specified: the initial condition and the conditions at domain boundaries. The initial conditions describe the domain of interest at time $t = t_0$ in the domain Ω while the conditions at domain boundaries describe the characteristics of the walls and specific domain boundaries.

The initial velocity fields are specified as

$$v_i(x_i, t=0) = V_i^0(x_i) \qquad \text{in } \Omega \tag{3.17}$$

There are two types of boundary conditions (Huang, 1999):

(a) Dirichlet essential boundary conditions

These boundary conditions are specified velocities at the boundaries. They can either be constant or vary with time.

$$v_i = f(x, y, z, t) \qquad \text{at boundary } S_v \tag{3.18}$$

Pressure is not specified as a boundary condition as it is an implicit variable in an incompressible flow that 'adjusts' itself to deliver the velocity field (Gresho *et al.*, 1980; Huang, 1999). However, in the case of contained flow in which velocities are specific on all boundaries, the pressure becomes indeterminate and it must be specified at least at one point as a datum.

(b) Neumann or natural boundary conditions

In this type of boundary condition, the normal and tangential traction forces may be specified for boundary S_f as below

$$f_n = -p + 2\mu \frac{\partial v_n}{\partial \hat{n}} \tag{3.19a}$$

$$f_t = \mu \left(\frac{\partial v_n}{\partial \hat{t}} + \frac{\partial v_t}{\partial \hat{n}} \right) \tag{3.19b}$$

where $\hat{n}$ and $\hat{t}$ are the unit normal and tangent vectors with respect to the boundary S_f. S_v and S_f are elements of the boundary S of the computational domain in a way that the following relationship holds,

$$S_v \cup S_f = S \tag{3.20a}$$

$$S_v \cap S_f = \phi \tag{3.20b}$$

where ϕ is the null set.

Non-Newtonian fluid mechanics often requires the stress components to be treated as dependent variables along with the velocity and pressure, a further complication from the classical Newtonian fluid situation. In the present study, the described governing equations are solved in conjunction with the following boundary conditions.

3.4.1 Inlet Boundary Condition

At the inlet, Dirichlet type uniform velocity are given with v_x, v_y and v_z. For the test cases that will be described in detail in Chapter 5, the shell is prescribed as non-slip non-permeable solid walls and the velocity components on these surfaces are equal to zero ($v_x = v_y = v_z = 0$).

3.4.2 Outlet Boundary conditions

Special care is needed in prescribing boundary conditions at the exit as earlier researchers such as Nassehi (1998) and more recently Das *et al.* (2002) have suggested that imposition of artificial exit boundary conditions might lead to unrealistic numerical results in simulations.

Exit conditions are usually unknowns and the subject of interest in computational modelling. Therefore, prescription of velocity at exit condition is not generally attempted. In the present study, only pressure boundary conditions are imposed at the exit, consistent with the explanation given in section 3.4 (a) and simulations showed that this give accurate results for the velocity components and the pressure.

3.5 Closure

In this chapter, the governing equations used in this study have been discussed. The challenge in the continuity equation due to the restrictions placed by the stability condition has been explained. This challenge has been overcome by adapting a slightly perturbed form of the continuity equation in this study. The assumptions and boundary conditions were also discussed. The equations are now ready to be discretized. The derivation of the working equations will be discussed in the next chapter.

CHAPTER 4

DERIVATION OF WORKING EQUATIONS

4.1 Mathematical Modelling Strategies

Many practical engineering problems described by a set of differential equations are either extremely difficult or impossible to be solved by conventional analytical methods. In the past, it was a common practice to simplify such problems to the point where an analytical solution could be obtained. With the recent advances in high-speed computer technology, the emphasis in engineering analysis has moved towards versatile computational modelling. At the core of every computational analysis is a numerical method that determines its accuracy, reliability, speed and computational cost. There are several numerical analysis techniques commonly employed by engineers to solve the non-linear governing partial differential equations (PDEs). Among these techniques are the *Finite Difference Method, Finite Volume Method, Finite Element Method* and the *Boundary Element Method.* These methods are briefly explained in this section.

The main idea of *Finite Difference Method* (FDM) is the discretization of the PDEs to reduce them to a set of algebraic equations. This is achieved by giving a pointwise approximation to the governing equations and the model is formed by writing difference equations for an array of grid points. The relations between the variables on the adjacent grid points are obtained by the Taylor series expansion, and the truncation error is given by the remainder of this series. This model can be notably improved when more points are used. In general, finite difference techniques can be used to treat fairly difficult problems, however for problems with irregular geometries or an unusual specification of boundary conditions, this technique becomes difficult to use and therefore unsuitable.

The *Finite Volume Method* (FVM) was actually derived from the FDM, but it subdivides the continuum domain into a discrete number of volumes, or cells. Each control volume is identified by the geometrical coordinates of the centre of cell and by the interfaces that separate one cell from the other. This method discretizes the integral form of the governing equations over each control volume, enforcing the conservation of the main conservative quantities, but it requires the approximation of the flux of transported quantities at the interfaces.

In the *Finite Element Method* (FEM), the solution domain can be analytically modelled or approximated by subdividing the continuum into discrete elements of any shape and size to give a piecewise approximation to the governing equations. And since these elements can be put together in a variety of ways, they can be used to represent highly complex and irregular shapes. Within each element some points are defined inside the element or on its sides. These points are the locations where the numerical values of the unknown variables should be determined. As far as the variables are concerned, they are approximated as a linear combination of chosen base functions.

Boundary Element Method (BEM), also known as *Boundary Integral Equation Method* (BIEM), uses Green's theorem to reduce the dimensionality of the problem in which a volume problem is reduced to a surface problem, and a surface problem is reduced to a line problem. The solution domain would have no interior mesh, but a mesh of connected points along the exterior boundary and a mesh of connected points along the interior boundary. The singular distributions always involve difficult integration over a point of singularity and special procedures need to be used for numerical implementation (Zienkiewicz and Taylor, 2000). This method is computationally less efficient and is not widely used in industry. It is sometimes combined with 'standard' finite element and has been occasionally described (Zienkiewicz and Taylor, 2000; Huebner *et al.*, 2001).

All these methods have strengths and weaknesses and a number of factors should be considered before deciding in favour of a particular method in modelling the problem at hand. These factors can be listed as follows:

(i) Type of governing equations of the process

(ii) Geometry of the process domain

(iii) Nature of the boundary conditions

(iv) Required accuracy of the calculations

(v) Computational cost

In solving non-linear field problems of geometrically complex domain under various types of boundary conditions, Finite Element Method was found to be the most appropriate technique for its flexibility and capability.

4.2 Finite Element Method

The finite element method is an approximate method for solving differential equations utilizing a variational principle and piecewise polynomial approximation. The mathematical concept of FEM can be traced back to 17th century where early researchers such as G.W. Leibnitz (1646-1716), Euler (1707-1783) utilized variational methods with the approximation approach in solving their mathematical equations such as the derivation of Euler equation. The first engineering application of FEM began in 1952 with an effort by Boeing to analyze aircraft structures in which a procedure was developed and appeared later in literature (Turner *et al.*, 1956). One of the co-authors, R.W. Clough, seems to be the first researcher to coin and use the term *finite element* in a paper published in 1960 (see Clough, 1960). A book by Argyris (1960) on energy theorems and matrix methods, alongside the first book on FEM published later by Zienkiewicz and Cheung (1967) stimulated further development in promoting the FEM. A good account of the FEM history can be found at Zienkiewicz and Taylor (2000), Babuŝka and Strouboulis (2001), Heinrich and Pepper (1999) and Huebner *et al.* (2001).

In recent years, finite element has found increasing application and wider acceptance in the application of general fluid mechanics, viscous fluid flow and heat transfer problems. The significant characteristic of geometrical flexibility made this technique a method of choice in problems posed in geometrically complex domains. For this reason, the analysis of industrial polymer and viscous processing flow regimes is often based on the finite element technique.

In finite element methods, there are a variety of ways in which the properties of the individual elements can be formulated. The most popularly used approaches are the direct approach, variational approach and the weighted residuals approach (Huebner *et al.*, 2001). Among these, the most versatile approach is the weighted residual approach which begins with the governing equations of the problem and proceeds without relying on a variational statement. This approach is advantageous as it becomes possible to extend the finite element method to problems where no functional is available and it is widely used to derive element properties for non-structural applications such as fluid mechanics. Therefore, a weighted residual approach will be used in conjunction with the standard Galerkin finite-element scheme in this study to generate a robust and practical numerical solution to the described problem domain.

In a continuum problem of any dimension, the field variable possesses infinitely many values because it is a function of each generic point in the body or solution region. Consequently, the problem is one with an infinite number of unknowns. The finite element discretization procedure reduces the problem to one of a finite number of unknowns by dividing the solution region into elements and by expressing the unknown field variable in terms of assumed approximating functions within each element. The solution of a continuum problem by the finite element method follows an orderly procedure. The steps are summarized as follows;

1. *Discretize the continuum.* The first step is to divide the continuum or solution region into elements. A variety of shapes such as tetrahedron and rectangular prism for three-dimensional domain may be used in the solution region. In this study, the

element discretization is achieved via a commercial finite element mesh generation software COSMOS™ GEOSTAR.

2. *Select interpolation functions.* After the domain is discretized into elements, the next step is to assign nodes to each element and then choose the interpolation function to represent the variation of the field variable over the element. The field variable may be a scalar, a vector, or a higher-order tensor. There are many types of functions such as trigonometric and polynomial functions that could be used in finite element analysis. However, polynomial functions are most widely used because they are relatively easy to manipulate mathematically and hence easy to integrate and differentiate. In this study, an eight-noded linear element of the Lagrange rectangular prisms family is used.

3. *Find the element properties.* Once the finite element and their interpolation functions have been selected, the matrix equations expressing the properties of the individual elements are determined. As discussed earlier, different approaches can be used to formulate the properties of the individual elements. The weighted residual approach is employed in this study for its versatile capability in coping with fluid mechanics applications.

4. *Assemble the element properties to obtain the system equations.* In order to find the properties of the overall systems modelled by the network of elements, all the element properties has to be 'assembled'. To achieve this, matrix equations expressing the behaviour of each element are combined to form the matrix equations that express the behaviour of the entire system. The matrix equations for the system have the same form as the equations for an individual element except that they contain many more terms because they include all nodes. The basis for the assembly procedure stems from the fact that at a node, where elements are interconnected, the

value of the field variable is the same for each element sharing that node. This becomes a unique feature of the finite element method where system equations are generated by the assembly of the individual element equations.

5. *Impose the boundary condition.* For the system equations to be solved, the appropriate boundary conditions relating to the problem must be specified. At this stage, known nodal values of the dependent variables are imposed to modify the systems equations accordingly. The required number of specified nodal variables is dictated by the physics of the problem and the complexity of the solution domain.

6. *Solve the system equations.* The assembly process resulted in a set of simultaneous equations that is to be solved to obtain the solutions for the defined problem. If the problem describes steady or equilibrium behaviour, then a set of linear or non-linear algebraic equations has to be solved. If the problem is unsteady, the nodal unknowns are a function of time, and a set of linear or non-linear ordinary differential equations should be solved.

7. *Make additional computations.* The solution of the system equations could be used to calculate other important parameters. For instance, element strains and stresses can be calculated from the displacements.

4.3 Choice of Interpolation Functions

A subject of utmost importance in finite element analysis is the selection of particular finite elements and the definitions of the appropriate interpolation functions within each element. The approximation functions, or interpolation functions, are defined in terms of the values of the field variables at specified points called nodes or nodal points. As such, the nature of the solution and the degree of approximation depend not only on the size and number of the elements used but also on the interpolation functions selected.

A general requirement in most finite element discretization is to maintain the compatibility of field variables across the boundaries of the neighbouring elements. Finite elements that generate uniquely defined function approximations along their sides or boundaries satisfy this condition. Elements that maintain inter-element compatibility of functions are called 'conforming elements'. The order of continuity of a conforming element that only ensures the compatibility of functions across its boundaries is said to be C^0. For problems requiring C^0 continuity, usually the simplest type of element is used to avoid excessive computational labour and therefore cost (Huebner *et al.*, 2001).

Commonly used C^0 three-dimensional elements include tetrahedron and rectangular prisms elements. Although tetrahedron element can effectively fill a complex three-dimensional region, it has proven a tedious affair to carry out manual mesh generation and data preparation. A more practical eight-noded linear rectangular prism element is chosen in this study as the master element, as shown in Figure 4.1. This element being the simplest member in the serendipity family of elements has been selected primarily for computing economy purposes although the higher order of the element family such as quadratic element (20 nodes) or cubic element (32 nodes) can be used in mesh refinement exercise. This is however beyond the scope of the present study. The shape functions are formulated as the products of Lagrange polynomials in the *x*-, *y*- and *z*- directions and the equations relating the Cartesian coordinates and the natural coordinates are

$$x = \sum_{i=1}^{8} x_i L_i \; ; \; y = \sum_{i=1}^{8} y_i L_i \; ; \; z = \sum_{i=1}^{8} z_i L_i \tag{4.1}$$

where

$$L_i = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i),\; i = 1,2,\ldots,8. \tag{4.2}$$

In Equation (4.2), (ξ_i, η_i, ζ_i) represents the coordinates of the node i in the (ξ,, η, ζ) natural coordinate system. More details on the natural coordinate systems of this element are elaborated in *Appendix A1*. Field variables such as velocity components and pressure are approximated using equal order interpolation functions. This corresponds to a total of 32 degrees of freedom consisting of 24 nodal velocity components and 8 nodal pressures.

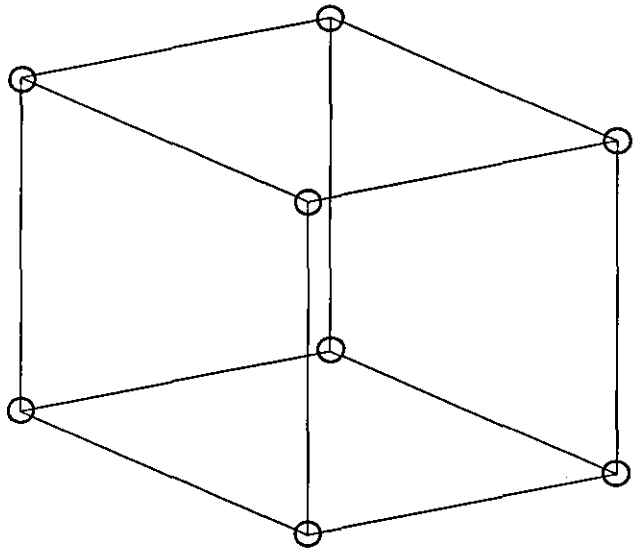


Figure 4.1 The linear prism element.

4.4 Inf-Sup Condition (Infinum-Supremum Condition)

For the analysis of fluid flows, mixed finite element procedures are generally used because the governing differential equations involve multiple field variables such as velocities and pressure. The displacement-based finite element method are relatively straightforward with the overall effectiveness of a discretization scheme can be demonstrated by a few test cases. A mixed method, however, may work remarkably well in the solution of some problems, and totally fail in other problems (Bathe, 2001). Although mixed finite element is based on a proper variational formulation, it does not ensure that the finite element method is reliable and that the method can be recommended for general use (Bathe, 1996). The necessity of a patch test has been briefly discussed in section 3.1.1 and will be further elaborated in this section.

It is crucial that finite element discretization, in whichever way the discretization has been formulated, be analyzed for its mathematical convergence characteristics. The specific conditions to be fulfilled are the consistency, ellipticity, and infinum-supremum conditions, which is also known as inf-sup conditions (Bathe, 2001). The fact that these conditions are satisfied ensured that the finite element discretizations are stable and moreover optimal. While the consistency and ellipticity conditions are fulfilled relatively easily, the applicable inf-sup condition is frequently difficult to satisfy.

The basic requirement that leads up to the definition of inf-sup has been given by Bathe (1996). The detailed mathematics will not be discussed in detail in this section as it is beyond the scope of the present study. The concept is however summarized in this section for the benefit of reference. The 'distance' between the exact solution **u** and the finite element space V_h is defined as

$$d(u,V_h) = \inf_{v_h \in V_h} \|u - v_h\| = \|u - \tilde{u}_h\| \tag{4.3}$$

where $\tilde{u}_h$ is an element in V_h but is in general not the finite element solution.

The purpose is to find conditions on V_h such that

$$\|u - \tilde{u}_h\| \le cd(u,V_h) \tag{4.4}$$

with a constant c independent of h and bulk modulus κ.

The inf-sup condition, which when satisfied ensures that Eq. (2) holds, is given as

$$\inf_{q_h \in D_h} \sup_{v_h \in V_h} \frac{\int_{Vol} q_h div v_h dVol}{\|v_h\|\|q_h\|} \ge \beta > 0 \tag{4.5}$$

with β a constant independent of h and κ.

Another useful form of the inf-sup condition is given as follows. For all **u** there is a $\mathbf{u}_I \in V_h$ (a vector that interpolates **u**) such that

$$\int_{Vol} div(u - u_I) q_h dVol = 0 \tag{4.6}$$

for all $q_h \in D_h$ where

$$\|u_I\| \le c\|u\|$$

with the constant c independent of **u**, $\mathbf{u}_I$ and h.

For incompressible and slightly compressible case, the inf-sup condition is frequently referred to as the Ladyzhenskaya-Babuŝka-Brezzi (LBB) condition because of the seminal papers by Ladyzhenskaya (1969), Babuŝka (1971) and Brezzi

(1974). Whether the inf-sup condition is satisfied depends on the following (Bathe, 1996):

(1) specific finite element used

(2) mesh topology

(3) boundary conditions

If the inf-sup values for these discretizations do not show a decrease towards zero, the test is passed, provided that there are also no spurious pressure modes (Bathe, 1996; 2001). If a discretization using a specific finite element always satisfies a given mathematical equation, for any mesh topology and boundary conditions, the element is said to satisfy the inf-sup condition. If, on the other hand, it is known that one mesh topology and/or one set of (physically realistic) boundary conditions for which the discretization does not satisfy the equation, then that element is simply said to not satisfy the inf-sup condition.

The results of analytical studies of the inf-sup characteristics of various displacement/pressure elements were summarized by Bathe (1996). However, an analytical proof of whether the inf-sup condition is satisfied by a specific element can be difficult, and for this reason a numerical test is valuable.

A widely used numerical inf-sup test has been proposed by Chapelle and Bathe (1993). Such a test can be applied to newly proposed elements and also to discretizations with elements of distorted geometries (analytical studies assumed homogeneous meshes of square elements). Discretizations based on distorted element meshes, which are virtually always used in engineering practice, and also some element discretizations, could not yet be proven analytically to satisfy the condition (Bathe, 2001). Thus, this numerical test could prove to be very useful.

While a numerical test cannot be as encompassing as an analytical proof, Bathe *et al.* (2000) suggested that when the numerical test is passed, in fact, the inf-sup condition is satisfied. Bathe (2001) also suggested that if the applicable inf-sup condition is satisfied (with the consistency and ellipticity requirement are fulfilled as well), the finite element scheme is reliable and effective and never 'fails'. However, he also stressed that the inf-sup condition is a severe requirement and a method not fulfilling the condition might still be acceptable.

Bathe *et al.* (2000) conducted an inf-sup test for distorted meshes in plate bending and they found that nine-node displacement-based quadrilateral element passes the inf-sup test for plate bending when uniform meshes with non-distorted elements are considered but fails when mesh distortions are introduced. This is consistent with the results reported earlier by Iosilevich *et al.* (1996, 1997).

For slightly compressible conditions, Bathe (1996) has analyzed theoretically and by use of numerical experiments *u/p* formation for 9/1, 9/3 (Crouzeix and Raviart) and 9/4 (Taylor-Hood) element. His studies showed that 9/1 element does not lock (i.e. finite element formulation which gives essentially the same accuracy in results for a given mesh irrespective of what Poisson's ratio is used, even when ν is close to 0.5), but the rate of convergence of pressure (hence stress) as the mesh is refined is only of $o(h)$ because a constant pressure is assumed in each nine-node element. The poor quality of the pressure prediction can have a negative effect on the prediction of the displacement displacement.

9/3 (Crouzeix and Raviart) was found to be most attractive because it does not lock and the stress convergence is of $o(h^2)$. Hence, the predictive capability is optimal since if a biquadratic displacement expansion is used, no higher-order convergence in stress can be expected.

For that reason, Bathe (1996) reasoned that many may be tempted to always use the 9/3 element. However, in practice, the 9/3 element is computationally slightly more expensive than the nine-node displacement-based element, and when ν is less than 0.48, the additional terms in the pressure expansion of the displacement-based element allow a slightly better prediction of stresses.

For 9/4 element (Taylor-Hood), the same studies showed that this element locks when ν is close to 0.50; hence it cannot be recommended for almost incompressible analysis. While the four-node 2-D and eight-node 3-D elements are extensively used in practice, the nine-node 2-D and 27-node 3-D elements are frequently more powerful.

As the detailed mathematics and numerical test for inf-sup condition is beyond the scope of the present study, further examples of analytical evaluation of inf-sup conditions can be referred in Bathe (1996) and numerical test at Babuŝka & Narasimhan (1997) and Babuŝka *et al.* (2002).

It should be noted that most study on inf-sup condition of distorted mesh were focused on 2-D elements and limited work has been conducted to study the effect of mesh distortion in 3-D elements. As attempts on 3-D modelling involving complex geometry have increased markedly in recent years, it may be necessary to investigate the effect of the inf-sup condition in distorted 3-D elements. In any case, it will be a major contribution towards the development of finite element applications, to which the scale of its study will involve further research and is not within the scope of the present study.

4.5 Solution Schemes of Stokes Equation

Application of the weighted residual method to the solution of the equations of continuity and motion can be based on a variety of schemes and techniques. The section that follows described general outlines and the formulation of the working equations of *Mixed Finite Element* U-V-W-P scheme (Zienkiewicz and Taylor, 2000) and *continuous Penalty* technique (Nassehi, 2002). In these formulations, the Stokes flow equation describing low Reynolds number flow regime is considered, therefore eliminating the convection term and the body force term in the motion equation. This has been discussed earlier in section 3.1.2.

4.5.1 U-V-W-P Scheme Based on Perturbed Continuity Method

As already explained (section 3.1.1.), the necessity to satisfy the LBB stability condition restricts the types of finite elements in the modelling of incompressible flow problems by the U-V-W-P scheme. To eliminate this restriction, the continuity equation representing the incompressible flow is replaced by an equation corresponding to slightly compressible fluid. Recapping the governing equations of continuity and motion given earlier in Equations (3.2), (3.8a), (3.8b) and (3.8c) as

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t}+\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \tag{3.2}$$

$$\rho\frac{\partial v_x}{\partial t}=-\frac{\partial p}{\partial x}+\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z} \tag{3.8a}$$

$$\rho\frac{\partial v_y}{\partial t}=-\frac{\partial p}{\partial y}+\frac{\partial \tau_{yx}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z} \tag{3.8b}$$

$$\rho\frac{\partial v_z}{\partial t}=-\frac{\partial p}{\partial z}+\frac{\partial \tau_{zx}}{\partial x}+\frac{\partial \tau_{zy}}{\partial y}+\frac{\partial \tau_{zz}}{\partial z} \tag{3.8c}$$

The inclusion of $(1/\rho c^2)(\partial p/\partial t)$ term in the continuity equation meant the transient terms should also be included in the Stokes equations. This inherently implied the use of a time stepping scheme is necessary in the solution of the equations. In this study, first-order Taylor-Galerkin time technique is used in conjunction with the U-V-W-P scheme. The Taylor-Galerkin method was first proposed for convective transport problems (Donea, 1984) and then applied to compressible flows (Löhner *et al.*, 1984a; Löhner *et al.*, 1984b; Bey *et al.*, 1985). The basic concept of Taylor-Galerkin is to use Taylor series expansion in time to establish recurrence relations for time marching and to use the method of weighted residuals with Galerkin's criteria to develop the finite element matrix equations describing the spatial distribution of the flow variables (Huebner *et al.*, 2001). A detailed description of this method can be found in Zienkiewicz and Taylor (2000), Huebner *et al.* (2001), Nassehi (2002) and Huang *et al.* (1999).

The solution domain is discretized into a mesh of finite element using the following expression,

$$\sum_{e=1}^{E}\Omega_e=\Omega \tag{4.7}$$

where E is the total number of elements. Taylor series expansion of the field variables with respect to the time increment at time step $n+\alpha\Delta t$ with $0 \leq \alpha \leq 1$ gives (Nassehi, 2002),

$$\frac{\Delta p}{\Delta t}=\frac{p|_{n+1}-p|_n}{\Delta t}=\left.\frac{\partial p}{\partial t}\right|_{n+\alpha\Delta t}+\frac{1}{2}\alpha\Delta t\left.\frac{\partial^2 p}{\partial t^2}\right|_{n+\alpha\Delta t} \tag{4.8a}$$

$$\frac{\Delta v_x}{\Delta t}=\frac{v_x|_{n+1}-v_x|_n}{\Delta t}=\left.\frac{\partial v_x}{\partial t}\right|_{n+\alpha\Delta t}+\frac{1}{2}\alpha\Delta t\left.\frac{\partial^2 v_x}{\partial t^2}\right|_{n+\alpha\Delta t} \tag{4.8b}$$

$$\frac{\Delta v_y}{\Delta t}=\frac{v_y|_{n+1}-v_y|_n}{\Delta t}=\left.\frac{\partial v_y}{\partial t}\right|_{n+\alpha\Delta t}+\frac{1}{2}\alpha\Delta t\left.\frac{\partial^2 v_y}{\partial t^2}\right|_{n+\alpha\Delta t} \tag{4.8c}$$

$$\frac{\Delta v_z}{\Delta t}=\frac{v_z|_{n+1}-v_z|_n}{\Delta t}=\left.\frac{\partial v_z}{\partial t}\right|_{n+\alpha\Delta t}+\frac{1}{2}\alpha\Delta t\left.\frac{\partial^2 v_z}{\partial t^2}\right|_{n+\alpha\Delta t} \tag{4.8d}$$

In practice, Taylor series expansion of the field variable is truncated and usually only the first few terms are kept. Although the accuracy of the time-stepping scheme in Taylor-Galerkin is dependent on the highest order of the time derivative remaining in the expansion after truncation, repeated differentiation and substitution of the temporal derivatives using the governing differential equations of complex field problem may prove to be overwhelming. The results obtained in this study showed accuracy is not compromised and therefore justified the omission of second order derivatives of the field variables. First order term derivatives can be found from Equations (3.2), (3.8a), (3.8b), (3.8c);

$$\left.\frac{\partial p}{\partial t}\right|_{n+\alpha\Delta t}=-\rho c^2\left.\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right|_{n+\alpha\Delta t} \tag{4.9a}$$

$$\left.\frac{\partial v_x}{\partial t}\right|_{n+\alpha\Delta t}=\frac{1}{\rho}\left.\left(-\frac{\partial p}{\partial x}+\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}\right)\right|_{n+\alpha\Delta t} \tag{4.9b}$$

$$\left.\frac{\partial v_y}{\partial t}\right|_{n+\alpha\Delta t}=\frac{1}{\rho}\left.\left(-\frac{\partial p}{\partial y}+\frac{\partial \tau_{yx}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}\right)\right|_{n+\alpha\Delta t} \tag{4.9c}$$

$$\left.\frac{\partial v_z}{\partial t}\right|_{n+\alpha\Delta t}=\frac{1}{\rho}\left.\left(-\frac{\partial p}{\partial z}+\frac{\partial \tau_{zx}}{\partial x}+\frac{\partial \tau_{zy}}{\partial y}+\frac{\partial \tau_{zz}}{\partial z}\right)\right|_{n+\alpha\Delta t} \tag{4.9d}$$

Second order term derivatives can be found by differentiating the first order terms. From Equation (4.8a);

$$\left.\frac{\partial^2 p}{\partial t^2}\right|_{n+\alpha\Delta t}=\frac{\partial}{\partial t}\left.\left(\frac{\partial p}{\partial t}\right)\right|_{n+\alpha\Delta t}$$

$$=\frac{\partial}{\partial t}\left.\left[-\rho c^2\frac{\partial}{\partial x}\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right]\right|_{n+\alpha\Delta t} \tag{4.10}$$

Applying the rules of integration yields

$$\left.\frac{\partial^2 p}{\partial t^2}\right|_{n+\alpha\Delta t}=\left.\left[\rho c^2\frac{\partial}{\partial x}\left(-\frac{\partial v_x}{\partial t}\right)-\rho c^2\frac{\partial}{\partial y}\left(-\frac{\partial v_y}{\partial t}\right)-\rho c^2\frac{\partial}{\partial z}\left(-\frac{\partial v_z}{\partial t}\right)\right]\right|_{n+\alpha\Delta t}$$

$$=\left.\left[\rho c^2\frac{\partial}{\partial x}\left(\frac{\partial p}{\partial x}-\frac{\partial \tau_{xx}}{\partial x}-\frac{\partial \tau_{xy}}{\partial y}-\frac{\partial \tau_{xz}}{\partial z}\right)\right]\right|_{n+\alpha\Delta t}+\left.\left[\rho c^2\frac{\partial}{\partial x}\left(\frac{\partial p}{\partial y}-\frac{\partial \tau_{yx}}{\partial x}-\frac{\partial \tau_{yy}}{\partial y}-\frac{\partial \tau_{yz}}{\partial z}\right)\right]\right|_{n+\alpha\Delta t}$$

$$+\left.\left[\rho c^2\frac{\partial}{\partial x}\left(\frac{\partial p}{\partial z}-\frac{\partial \tau_{zx}}{\partial x}-\frac{\partial \tau_{zy}}{\partial y}-\frac{\partial \tau_{zz}}{\partial z}\right)\right]\right|_{n+\alpha\Delta t} \tag{4.11}$$

Terms that contain τ_{xx}, τ_{xy}, τ_{xz}, τ_{yx}, τ_{yy}, τ_{yz}, τ_{zy}, τ_{zz} are ignored since the order of derivatives are higher than those of other terms. Thus, we have

$$\left.\frac{\partial^2 p}{\partial t^2}\right|_{n+\alpha\Delta t}=\left.\left[\rho c^2\frac{\partial^2 p}{\partial x^2}+\rho c^2\frac{\partial^2 p}{\partial y^2}+\rho c^2\frac{\partial^2 p}{\partial z^2}\right]\right|_{n+\alpha\Delta t} \tag{4.12}$$

From Equation (4.8b),

$$\left.\frac{\partial^2 v_x}{\partial t^2}\right|_{n+\alpha\Delta t}=\frac{\partial}{\partial t}\left.\left(\frac{\partial v_x}{\partial t}\right)\right|_{n+\alpha\Delta t}=\frac{\partial}{\partial t}\left.\left(-\frac{1}{\rho}\frac{\partial p}{\partial x}\right)\right|_{n+\alpha\Delta t}$$

$$+\frac{1}{\rho}\frac{\partial}{\partial t}\left.\left(\frac{\partial \tau_{xx}}{\partial x}\right)\right|_{n+\alpha\Delta t}+\frac{1}{\rho}\frac{\partial}{\partial t}\left.\left(\frac{\partial \tau_{xy}}{\partial y}\right)\right|_{n+\alpha\Delta t}+\frac{1}{\rho}\frac{\partial}{\partial t}\left.\left(\frac{\partial \tau_{xz}}{\partial z}\right)\right|_{n+\alpha\Delta t} \tag{4.13}$$

The second terms onwards at the right hand side of equation were ignored since the order of derivatives is higher than that of first term. Applying the rules of differentiation to Equations (4.13), and similarly to the second order term in Equations (4.8c) and (4.8d) yields

$$\left.\frac{\partial^2 v_x}{\partial t^2}\right|_{n+\alpha\Delta t} = -\frac{1}{\rho}\frac{\partial}{\partial x}\left.\left(\frac{\partial p}{\partial t}\right)\right|_{n+\alpha\Delta t} = c^2\frac{\partial}{\partial x}\left.\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right|_{n+\alpha\Delta t} \qquad (4.14a)$$

$$\left.\frac{\partial^2 v_y}{\partial t^2}\right|_{n+\alpha\Delta t} = -\frac{1}{\rho}\frac{\partial}{\partial y}\left.\left(\frac{\partial p}{\partial t}\right)\right|_{n+\alpha\Delta t} = c^2\frac{\partial}{\partial y}\left.\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right|_{n+\alpha\Delta t} \qquad (4.14b)$$

$$\left.\frac{\partial^2 v_z}{\partial t^2}\right|_{n+\alpha\Delta t} = -\frac{1}{\rho}\frac{\partial}{\partial z}\left.\left(\frac{\partial p}{\partial t}\right)\right|_{n+\alpha\Delta t} = c^2\frac{\partial}{\partial z}\left.\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right|_{n+\alpha\Delta t} \qquad (4.14c)$$

Substituting the first order and second order terms from Equations (4.9a), (4.9b), (4.9c), (4.9d), and Equations (4.12), (4.14a), (4.14b), (4.14c) into Equations (4.8a), (4.8b), (4.8c), (4.8d) yields

$$\frac{p|_{n+1} - p|_n}{\Delta t} = -\rho c^2\left.\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t\rho c^2\left.\left(\frac{\partial^2 p}{\partial x^2}+\frac{\partial^2 p}{\partial y^2}+\frac{\partial^2 p}{\partial z^2}\right)\right|_{n+\alpha\Delta t} \qquad (4.15a)$$

$$\frac{v_x|_{n+1} - v_x|_n}{\Delta t} = -\frac{1}{\rho}\left.\frac{\partial p}{\partial x}\right|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\left.\left[2\frac{\partial^2 v_x}{\partial x^2}+\frac{\partial^2 v_x}{\partial y^2}+\frac{\partial}{\partial y}\left(\frac{\partial v_y}{\partial x}\right)+\frac{\partial}{\partial z}\left(\frac{\partial v_z}{\partial x}\right)+\frac{\partial^2 v_x}{\partial z^2}\right]\right|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t\left.\left[c^2\frac{\partial}{\partial x}\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)\right]\right|_{n+\alpha\Delta t} \qquad (4.15b)$$

$$\frac{v_y\big|_{n+1} - v_y\big|_n}{\Delta t} = -\frac{1}{\rho}\frac{\partial p}{\partial y}\bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\left[\frac{\partial}{\partial x}\left(\frac{\partial v_x}{\partial y}\right) + \frac{\partial^2 v_y}{\partial x^2} + 2\frac{\partial^2 v_y}{\partial y^2} + \frac{\partial^2 v_y}{\partial z^2} + \frac{\partial}{\partial z}\left(\frac{\partial v_z}{\partial y}\right)\right]\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t\left[c^2\frac{\partial}{\partial y}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right)\right]\Bigg|_{n+\alpha\Delta t} \qquad (4.15c)$$

$$\frac{v_z\big|_{n+1} - v_z\big|_n}{\Delta t} = -\frac{1}{\rho}\frac{\partial p}{\partial z}\bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\left[\frac{\partial^2 v_x}{\partial z^2} + \frac{\partial}{\partial x}\left(\frac{\partial v_x}{\partial z}\right) + \frac{\partial}{\partial y}\left(\frac{\partial v_y}{\partial z}\right) + \frac{\partial^2 v_z}{\partial y^2} + 2\frac{\partial^2 v_z}{\partial z^2}\right]\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t\left[c^2\frac{\partial}{\partial z}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right)\right]\Bigg|_{n+\alpha\Delta t} \qquad (4.15d)$$

In the *Mixed Finite Element* U-V-W-P technique, both velocity and pressure in the governing equations are regarded as primary variables and are discretized as unknowns. Approximation of the unknown variables v_x, v_y, v_z and p over an element gives

$$v_x = u \approx \tilde{u} = \sum_{j=1}^{n} N_j u_j\ ,\quad v_y = v \approx \tilde{v} = \sum_{j=1}^{n} N_j v_j\ ,$$

$$v_z = w \approx \tilde{w} = \sum_{j=1}^{n} N_j w_j\ ,\quad p \approx \tilde{p} = \sum_{j=1}^{n} N_j p_j \qquad (4.16a, b, c, d)$$

where N_j is the shape function associated with nodes $j = 1,\ldots..,n$ with n the number of nodes per element. Applying the standard Galerkin method to Equations (4.15a), (4.15b), (4.15c), (4.15d) and substituting Equations (4.16a, b, c, d) yields

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j p_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j p_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t}$$

$$= -\rho c^2 \int\limits_{\Omega_e} N_i \left(\frac{\partial \sum\limits_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum\limits_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum\limits_{j=1}^{n} N_j w_j}{\partial z} \right) d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

$$+ \frac{1}{2}\alpha\Delta t \rho c^2 \int\limits_{\Omega_e} N_i \left[\frac{\partial}{\partial x}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j p_j}{\partial x} \right) + \frac{\partial}{\partial y}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j p_j}{\partial y} \right) + \frac{\partial}{\partial z}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j p_j}{\partial z} \right) \right] d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

(4.17a)

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j u_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j u_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t} = -\frac{1}{\rho}\int\limits_{\Omega_e} N_i \frac{\partial}{\partial x}\left(\sum_{j=1}^{n} N_j p_j \right) d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

$$+ \frac{\eta}{\rho}\int\limits_{\Omega_e} N_i \left[2\frac{\partial}{\partial x}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j u_j}{\partial x} \right) + \frac{\partial}{\partial y}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j v_j}{\partial y} \right) + \frac{\partial}{\partial y}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j v_j}{\partial x} \right) + \frac{\partial}{\partial z}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j w_j}{\partial x} \right) + \frac{\partial}{\partial z}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j w_j}{\partial z} \right) \right] d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

$$+ \frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Omega_e} N_i \frac{\partial}{\partial x}\left(\frac{\partial \sum\limits_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum\limits_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum\limits_{j=1}^{n} N_j w_j}{\partial z} \right) d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

(4.17b)

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum\limits_{j=1}^{n} N_j v_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum\limits_{j=1}^{n} N_j v_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t} = -\frac{1}{\rho}\int\limits_{\Omega_e} N_i \frac{\partial}{\partial y}\left(\sum_{j=1}^{n} N_j p_j\right) d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\int\limits_{\Omega_e} N_i \left[\frac{\partial}{\partial x}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j u_j}{\partial y}\right)+\frac{\partial}{\partial x}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j v_j}{\partial x}\right)+2\frac{\partial}{\partial y}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j v_j}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j v_j}{\partial z}\right)+\frac{\partial}{\partial z}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j w_j}{\partial y}\right)\right] d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Omega_e} N_i \frac{\partial}{\partial y}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j u_j}{\partial x}+\frac{\partial\sum\limits_{j=1}^{n} N_j v_j}{\partial y}+\frac{\partial\sum\limits_{j=1}^{n} N_j w_j}{\partial z}\right) d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

(4.17c)

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum\limits_{j=1}^{n} N_j w_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum\limits_{j=1}^{n} N_j w_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t} = -\frac{1}{\rho}\int\limits_{\Omega_e} N_i \frac{\partial}{\partial z}\left(\sum_{j=1}^{n} N_j p_j\right) d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\int\limits_{\Omega_e} N_i \left[\frac{\partial}{\partial z}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j u_j}{\partial z}\right)+\frac{\partial}{\partial x}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j u_j}{\partial z}\right)+\frac{\partial}{\partial y}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j v_j}{\partial z}\right)+\frac{\partial}{\partial y}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j w_j}{\partial y}\right)+2\frac{\partial}{\partial z^2}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j w_j}{\partial z}\right)\right] d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Omega_e} N_i \frac{\partial}{\partial z}\left(\frac{\partial\sum\limits_{j=1}^{n} N_j u_j}{\partial x}+\frac{\partial\sum\limits_{j=1}^{n} N_j v_j}{\partial y}+\frac{\partial\sum\limits_{j=1}^{n} N_j w_j}{\partial z}\right) d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

(4.17d)

where N_i is the weight function.

At this stage, the formulated equations contain second-order derivatives. Before the system equations can be assembled from the element equations, it is required that the choice of approximating functions guarantees the inter-element continuity necessary for the assembly process. In this case, C^0 elements cannot generate an acceptable solution for the equation due to the fact that the first derivatives of the shape functions will be discontinuous across element boundaries and the integral of their second derivatives will tend to infinity (Nassehi, 2002; Huebner *et al.*, 2001). To overcome this difficulty, the second derivatives in Equations (4.17a), (4.17b), (4.17c) and (4.17d) are integrated by parts to obtain a 'weak' form of the equations. In this way, expressions containing lower-order derivatives are obtained hence permitting the use of approximating functions with lower-order inter-element continuity. When integration by parts is possible, it also offers a convenient way to introduce the natural boundary conditions that must be satisfied on some portion of the boundary. Although the boundary terms containing the natural boundary conditions appear in the equations for each element, in the assembly of the element equations only the boundary elements give non-vanishing contributions (Huebner *et al.*, 2001).

After integration by parts using Gauss' theorem (Huebner *et al.*, 2001), the Equations (4.17a), (4.17b), (4.17c) and (4.17d) becomes

$$
\frac{\left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j p_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j p_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t}
$$

$$
= -\rho c^2 \int\limits_{\Omega_e} N_i \left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z} \right) d\Omega_e \Bigg|_{n+\alpha\Delta t}
$$

$$
-\frac{1}{2}\alpha\Delta t \rho c^2 \int\limits_{\Omega_e} \left[\frac{\partial N_i}{\partial x} \left(\frac{\partial \sum_{j=1}^{n} N_j p_j}{\partial x} \right) + \frac{\partial N_i}{\partial y} \left(\frac{\partial \sum_{j=1}^{n} N_j p_j}{\partial y} \right) + \frac{\partial N_i}{\partial z} \left(\frac{\partial \sum_{j=1}^{n} N_j p_j}{\partial z} \right) \right] d\Omega_e \Bigg|_{n+\alpha\Delta t}
$$

$$
+\frac{1}{2}\alpha\Delta t \rho c^2 \int\limits_{\Gamma_e} \left[N_i \left(\frac{\partial \sum_{j=1}^{n} N_j p_j}{\partial x} \right) \hat{n}_x + N_i \left(\frac{\partial \sum_{j=1}^{n} N_j p_j}{\partial y} \right) \hat{n}_y + N_i \left(\frac{\partial \sum_{j=1}^{n} N_j p_j}{\partial z} \right) \hat{n}_z \right] d\Gamma_e \Bigg|_{n+\alpha\Delta t}
$$

(4.18a)

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j u_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j u_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t}$$

$$= \frac{1}{\rho}\int\limits_{\Omega_e} \frac{\partial N_i}{\partial x}\left(\sum_{j=1}^{n} N_j p_j\right) d\Omega_e\Bigg|_{n+\alpha\Delta t} - \frac{1}{\rho}\int\limits_{\Omega_e} N_i \tilde{p}_j \hat{n}_x d\Gamma_e\Bigg|_{n+\alpha\Delta t}$$

$$-\frac{\eta}{\rho}\int\limits_{\Omega_e}\left[2\frac{\partial N_i}{\partial x}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x}\right) + \frac{\partial N_i}{\partial y}\left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y}\right) + \frac{\partial N_i}{\partial y}\left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial x}\right) + \frac{\partial N_i}{\partial z}\left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial x}\right) + \frac{\partial N_i}{\partial z}\left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right)\right] d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\int\limits_{\Gamma_e} N_i\left[2\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x}\right)\hat{n}_x + \left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y}\right)\hat{n}_y + \left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial x}\right)\hat{n}_y + \left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial x}\right)\hat{n}_z + \left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right)\hat{n}_z\right] d\Gamma_e\Bigg|_{n+\alpha\Delta t}$$

$$-\frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Omega_e} \frac{\partial N_i}{\partial x}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right) d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Gamma_e} N_i\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right) \hat{n}_x d\Gamma_e\Bigg|_{n+\alpha\Delta t}$$

(4.18b)

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j v_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j v_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t}$$

$$= \frac{1}{\rho}\int\limits_{\Omega_e}\frac{\partial N_i}{\partial y}\left(\sum_{j=1}^{n} N_j p_j\right) d\Omega_e\Bigg|_{n+\alpha\Delta t} - \frac{1}{\rho}\int\limits_{\Gamma_e} N_i \tilde{p}_j \hat{n}_y d\Gamma_e\Bigg|_{n+\alpha\Delta t}$$

$$-\frac{\eta}{\rho}\int\limits_{\Omega_e}\frac{\partial N_i}{\partial x}\left(\frac{\partial\sum_{j=1}^{n} N_j u_j}{\partial y}\right)+\frac{\partial N_i}{\partial x}\left(\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial x}\right)+2\frac{\partial N_i}{\partial y}\left(\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial y}\right)+\frac{\partial N_i}{\partial z}\left(\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial z}\right)+\frac{\partial N_i}{\partial z}\left(\frac{\partial\sum_{j=1}^{n} N_j w_j}{\partial y}\right)d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\int\limits_{\Gamma_e} N_i\left[\left(\frac{\partial\sum_{j=1}^{n} N_j u_j}{\partial y}\right)\hat{n}_x+\left(\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial x}\right)\hat{n}_x+2\left(\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial y}\right)\hat{n}_y+\left(\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial z}\right)\hat{n}_z+\left(\frac{\partial\sum_{j=1}^{n} N_j w_j}{\partial y}\right)\hat{n}_z\right]d\Gamma_e\Bigg|_{n+\alpha\Delta t}$$

$$-\frac{1}{2}\alpha\Delta t c^2\int\limits_{\Omega_e}\frac{\partial N_i}{\partial y}\left(\frac{\partial\sum_{j=1}^{n} N_j u_j}{\partial x}+\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial y}+\frac{\partial\sum_{j=1}^{n} N_j w_j}{\partial z}\right)d\Omega_e\Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t c^2\int\limits_{\Gamma_e} N_i\left(\frac{\partial\sum_{j=1}^{n} N_j u_j}{\partial x}+\frac{\partial\sum_{j=1}^{n} N_j v_j}{\partial y}+\frac{\partial\sum_{j=1}^{n} N_j w_j}{\partial z}\right)\hat{n}_y d\Gamma_e\Bigg|_{n+\alpha\Delta t}$$

(4.18c)

$$\frac{\left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j w_j d\Omega_e\right\}\Bigg|_{n+1} - \left\{\int\limits_{\Omega_e} N_i \sum_{j=1}^{n} N_j w_j d\Omega_e\right\}\Bigg|_{n}}{\Delta t}$$

$$= \frac{1}{\rho}\int\limits_{\Omega_e} \frac{\partial N_i}{\partial z}\left(\sum_{j=1}^{n} N_j p_j\right) d\Omega_e \Bigg|_{n+\alpha\Delta t} - \frac{1}{\rho}\int\limits_{\Gamma_e} N_i \tilde{p}_j \hat{n}_z d\Gamma_e \Bigg|_{n+\alpha\Delta t}$$

$$-\frac{\eta}{\rho}\int\limits_{\Omega_e}\left[\frac{\partial N_i}{\partial z}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial z}\right) + \frac{\partial N_i}{\partial x}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial z}\right) + \frac{\partial N_i}{\partial y}\left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial z}\right) + \frac{\partial N_i}{\partial y}\left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial y}\right) + 2\frac{\partial N_i}{\partial z}\left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right)\right] d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

$$+\frac{\eta}{\rho}\int\limits_{\Gamma_e} N_i\left[\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial z}\right)\hat{n}_z + \left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial z}\right)\hat{n}_x + \left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial z}\right)\hat{n}_y + \left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial y}\right)\hat{n}_y + 2\left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right)\hat{n}_z\right] d\Gamma_e \Bigg|_{n+\alpha\Delta t}$$

$$-\frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Omega_e} \frac{\partial N_i}{\partial z}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right) d\Omega_e \Bigg|_{n+\alpha\Delta t}$$

$$+\frac{1}{2}\alpha\Delta t c^2 \int\limits_{\Gamma_e} N_i\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z}\right) \hat{n}_z d\Gamma_e \Bigg|_{n+\alpha\Delta t}$$

(4.18d)

The result of integration by parts is that the order of differentiation is reduced by one in the resulting equations. The continuity restrictions on the approximating function have been reduced or 'weakened'. The resulting integral equation is referred to as the 'weak form' of the boundary value problem. In these equations, functions given at time level $n+\alpha\Delta t$ can be interpolated as (Nassehi, 2002)

$$F\big|_{n+\alpha\Delta t} = \alpha F\big|_{n+1} + (1-\alpha)F\big|_{n} \tag{4.19}$$

The final working equations of this scheme for the Stokes flow model can be written as:

$$\begin{bmatrix} M^{11}{}_{ij} & M^{12}{}_{ij} & M^{13}{}_{ij} & M^{14}{}_{ij} \\ M^{21}{}_{ij} & M^{22}{}_{ij} & M^{23}{}_{ij} & M^{24}{}_{ij} \\ M^{31}{}_{ij} & M^{32}{}_{ij} & M^{33}{}_{ij} & M^{34}{}_{ij} \\ M^{41}{}_{ij} & M^{42}{}_{ij} & M^{43}{}_{ij} & M^{44}{}_{ij} \end{bmatrix}^{n+1} \begin{Bmatrix} u_j \\ v_j \\ w_j \\ p_j \end{Bmatrix}^{n+1} =$$

$$\begin{bmatrix} K^{11}{}_{ij} & K^{12}{}_{ij} & K^{13}{}_{ij} & K^{14}{}_{ij} \\ K^{21}{}_{ij} & K^{22}{}_{ij} & K^{23}{}_{ij} & K^{24}{}_{ij} \\ K^{31}{}_{ij} & K^{32}{}_{ij} & K^{33}{}_{ij} & K^{34}{}_{ij} \\ K^{41}{}_{ij} & K^{42}{}_{ij} & K^{43}{}_{ij} & K^{44}{}_{ij} \end{bmatrix}^{n} \begin{Bmatrix} u_j \\ v_j \\ w_j \\ p_j \end{Bmatrix}^{n} + \begin{Bmatrix} B^1{}_j \\ B^2{}_j \\ B^3{}_j \\ B^4{}_j \end{Bmatrix}^{n+1} + \begin{Bmatrix} C^1{}_j \\ C^2{}_j \\ C^3{}_j \\ C^4{}_j \end{Bmatrix}^{n} \tag{4.20}$$

where

$$M^{11}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{\alpha\Delta t}{\rho}\left[\left(2\eta + \frac{1}{2}\Delta t c^2\right)\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\} dxdydz$$

(4.21-1)

$$M^{12}{}_{ij} = \iiint_{\Omega_e} \left\{ \alpha\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x} + \frac{1}{2}\Delta t c^2 \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}\right)\right\} dxdydz \tag{4.21-2}$$

$$M^{13}{}_{ij} = \iiint_{\Omega_e} \left\{ \alpha\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x} + \frac{1}{2}\Delta t c^2 \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z}\right)\right\} dxdydz \tag{4.21-3}$$

$$M^{14}{}_{ij} = \iiint_{\Omega_e} \left\{ -\frac{\alpha\Delta t}{\rho}\frac{\partial N_i}{\partial x} N_j \right\} dxdydz \tag{4.21-4}$$

$$M^{21}{}_{ij} = \iiint_{\Omega_e} \left\{ \alpha\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y} + \frac{1}{2}\Delta t c^2 \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x}\right)\right\} dxdydz \tag{4.21-5}$$

$$M^{22}{}_{ij}=\iiint_{\Omega_e}\left\{N_iN_j+\alpha\Delta t\left[\frac{\eta}{\rho}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}+\left(\frac{2\eta}{\rho}+\frac{1}{2}\Delta tc^2\right)\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}+\eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\}dxdydz \tag{4.21-6}$$

$$M^{23}{}_{ij}=\iiint_{\Omega_e}\left\{\alpha\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial y}+\frac{1}{2}\Delta tc^2\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial z}\right)\right\}dxdydz \tag{4.21-7}$$

$$M^{24}{}_{ij}=\iiint_{\Omega_e}\left\{-\frac{\alpha\Delta t}{\rho}\frac{\partial N_i}{\partial y}N_j\right\}dxdydz \tag{4.21-8}$$

$$M^{31}{}_{ij}=\iiint_{\Omega_e}\left\{\alpha\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z}+\frac{1}{2}\Delta tc^2\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x}\right)\right\}dxdydz \tag{4.21-9}$$

$$M^{32}{}_{ij}=\iiint_{\Omega_e}\left\{\alpha\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial z}+\frac{1}{2}\Delta tc^2\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial y}\right)\right\}dxdydz \tag{4.21-10}$$

$$M^{33}{}_{ij}=\iiint_{\Omega_e}\left\{N_iN_j+\alpha\Delta t\left[\frac{\eta}{\rho}\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}+\frac{\eta}{\rho}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}+\left(\frac{2\eta}{\rho}+\frac{1}{2}\Delta tc^2\right)\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\}dxdydz \tag{4.21-11}$$

$$M^{34}{}_{ij}=\iiint_{\Omega_e}\left\{-\frac{\alpha\Delta t}{\rho}\frac{\partial N_i}{\partial z}N_j\right\}dxdydz \tag{4.21-12}$$

$$M^{41}{}_{ij}=\iiint_{\Omega_e}\left\{\rho\alpha\Delta tc^2N_i\frac{\partial N_j}{\partial x}\right\}dxdydz \tag{4.21-13}$$

$$M^{42}{}_{ij}=\iiint_{\Omega_e}\left\{\rho\alpha\Delta tc^2N_i\frac{\partial N_j}{\partial y}\right\}dxdydz \tag{4.21-14}$$

$$M^{43}{}_{ij}=\iiint_{\Omega_e}\left\{\rho\alpha\Delta tc^2N_i\frac{\partial N_j}{\partial z}\right\}dxdydz \tag{4.21-15}$$

$$M^{44}{}_{ij}=\iiint_{\Omega_e}\left\{N_iN_j+\rho\alpha\Delta t\frac{1}{2}\Delta tc^2\left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}+\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}+\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right)\right\}dxdydz \tag{4.21-16}$$

$$K^{11}{}_{ij}=\iiint_{\Omega_e}\left\{N_iN_j+\frac{(1-\alpha)\Delta t}{\rho}\left[\left(2\eta+\frac{1}{2}\Delta tc^2\right)\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}+\eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}+\eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\}dxdydz \tag{4.21-17}$$

$$K^{12}{}_{ij}=\iiint_{\Omega_e}\left\{(1-\alpha)\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x}+\frac{1}{2}\Delta tc^2\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}\right)\right\}dxdydz \tag{4.21-18}$$

$$K^{13}{}_{ij}=\iiint_{\Omega_e}\left\{(1-\alpha)\Delta t\left(\frac{\eta}{\rho}\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x}+\frac{1}{2}\Delta tc^2\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z}\right)\right\}dxdydz \tag{4.21-19}$$

$$K^{14}{}_{ij} = \iiint_{\Omega_e} \left\{ -\frac{(1-\alpha)\Delta t}{\rho} \frac{\partial N_i}{\partial x} N_j \right\} dxdydz \tag{4.21-20}$$

$$K^{21}{}_{ij} = \iiint_{\Omega_e} \left\{ (1-\alpha)\Delta t \left(\frac{\eta}{\rho} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial y} + \frac{1}{2}\Delta tc^2 \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial x} \right) \right\} dxdydz \tag{4.21-21}$$

$$K^{22}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + (1-\alpha)\Delta t \left[\frac{\eta}{\rho} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \left(\frac{2\eta}{\rho} + \frac{1}{2}\Delta tc^2 \right) \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \eta \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right] \right\} dxdydz \tag{4.21-22}$$

$$K^{23}{}_{ij} = \iiint_{\Omega_e} \left\{ (1-\alpha)\Delta t \left(\frac{\eta}{\rho} \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial y} + \frac{1}{2}\Delta tc^2 \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial z} \right) \right\} dxdydz \tag{4.21-23}$$

$$K^{24}{}_{ij} = \iiint_{\Omega_e} \left\{ -\frac{(1-\alpha)\Delta t}{\rho} \frac{\partial N_i}{\partial y} N_j \right\} dxdydz \tag{4.21-24}$$

$$K^{31}{}_{ij} = \iiint_{\Omega_e} \left\{ (1-\alpha)\Delta t \left(\frac{\eta}{\rho} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial z} + \frac{1}{2}\Delta tc^2 \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial x} \right) \right\} dxdydz \tag{4.21-25}$$

$$K^{32}{}_{ij} = \iiint_{\Omega_e} \left\{ (1-\alpha)\Delta t \left(\frac{\eta}{\rho} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial z} + \frac{1}{2}\Delta tc^2 \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial y} \right) \right\} dxdydz \tag{4.21-26}$$

$$K^{33}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + (1-\alpha)\Delta t \left[\frac{\eta}{\rho} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\eta}{\rho} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \left(\frac{2\eta}{\rho} + \frac{1}{2}\Delta tc^2 \right) \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right] \right\} dxdydz \tag{4.21-27}$$

$$K^{34}{}_{ij} = \iiint_{\Omega_e} \left\{ -\frac{(1-\alpha)\Delta t}{\rho} \frac{\partial N_i}{\partial z} N_j \right\} dxdydz \tag{4.21-28}$$

$$K^{41}{}_{ij} = \iiint_{\Omega_e} \left\{ \rho(1-\alpha)\Delta tc^2 N_i \frac{\partial N_j}{\partial x} \right\} dxdydz \tag{4.21-29}$$

$$K^{42}{}_{ij} = \iiint_{\Omega_e} \left\{ \rho(1-\alpha)\Delta tc^2 N_i \frac{\partial N_j}{\partial y} \right\} dxdydz \tag{4.21-30}$$

$$K^{43}{}_{ij} = \iiint_{\Omega_e} \left\{ \rho(1-\alpha)\Delta tc^2 N_i \frac{\partial N_j}{\partial z} \right\} dxdydz \tag{4.21-31}$$

$$K^{44}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \rho(1-\alpha)\Delta t \frac{1}{2}\Delta tc^2 \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right) \right\} dxdydz \tag{4.21-32}$$

$$B^1{}_j = \int_{\Gamma_e} \left\{ N_i \frac{\alpha\Delta t}{\rho} \left[\left(2\eta + \frac{1}{2}\Delta t c^2 \right) \frac{\partial \tilde{u}}{\partial x} n_x + \eta \frac{\partial \tilde{u}}{\partial y} n_y + \eta \frac{\partial \tilde{u}}{\partial z} n_z \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{v}}{\partial x} n_y + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{v}}{\partial y} n_z + \eta \frac{\partial \tilde{w}}{\partial x} n_z + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{w}}{\partial z} n_x - \tilde{p} n_x \right] \right\} d\Gamma_e \qquad (4.21\text{-}33)$$

$$B^2{}_j = \int_{\Gamma_e} \left\{ N_i \frac{\alpha\Delta t}{\rho} \left[\frac{1}{2}\Delta t c^2 \frac{\partial \tilde{u}}{\partial x} n_y + \eta \frac{\partial \tilde{u}}{\partial y} n_x + \eta \frac{\partial \tilde{v}}{\partial x} n_x + \left(2\eta + \frac{1}{2}\Delta t c^2 \right) \frac{\partial \tilde{v}}{\partial y} n_y \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{v}}{\partial z} n_z + \eta \frac{\partial \tilde{w}}{\partial y} n_z + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{w}}{\partial z} n_y - \tilde{p} n_y \right] \right\} d\Gamma_e \qquad (4.21\text{-}34)$$

$$B^3{}_j = \int_{\Gamma_e} \left\{ N_i \frac{\alpha\Delta t}{\rho} \left[\frac{1}{2}\Delta t c^2 \frac{\partial \tilde{u}}{\partial x} n_z + \eta \frac{\partial \tilde{u}}{\partial z} n_x + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{v}}{\partial y} n_z + \eta \frac{\partial \tilde{v}}{\partial z} n_y - \tilde{p} n_z \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial x} n_x + \eta \frac{\partial \tilde{w}}{\partial y} n_y + \left(2\eta + \frac{1}{2}\Delta t c^2 \right) \frac{\partial \tilde{w}}{\partial z} n_z \right] \right\} d\Gamma_e \qquad (4.21\text{-}35)$$

$$B^4{}_j = \int_{\Gamma_e} \left\{ N_i \rho\alpha\Delta t^2 \frac{1}{2} c^2 \left(\frac{\partial \tilde{p}}{\partial x} n_x + \frac{\partial \tilde{p}}{\partial y} n_y + \frac{\partial \tilde{p}}{\partial z} n_z \right) \right\} d\Gamma_e \qquad (4.21\text{-}36)$$

$$C^1{}_j = \int_{\Gamma_e} \left\{ N_i \frac{(1-\alpha)\Delta t}{\rho} \left[\left(2\eta + \frac{1}{2}\Delta t c^2 \right) \frac{\partial \tilde{u}}{\partial x} n_x + \eta \frac{\partial \tilde{u}}{\partial y} n_y + \eta \frac{\partial \tilde{u}}{\partial z} n_z \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{v}}{\partial x} n_y + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{v}}{\partial y} n_z + \eta \frac{\partial \tilde{w}}{\partial x} n_z + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{w}}{\partial z} n_x - \tilde{p} n_x \right] \right\} d\Gamma_e \qquad (4.21\text{-}37)$$

$$C^2{}_j = \int_{\Gamma_e} \left\{ N_i \frac{(1-\alpha)\Delta t}{\rho} \left[\frac{1}{2}\Delta t c^2 \frac{\partial \tilde{u}}{\partial x} n_y + \eta \frac{\partial \tilde{u}}{\partial y} n_x + \eta \frac{\partial \tilde{v}}{\partial x} n_x + \left(2\eta + \frac{1}{2}\Delta t c^2 \right) \frac{\partial \tilde{v}}{\partial y} n_y \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{v}}{\partial z} n_z + \eta \frac{\partial \tilde{w}}{\partial y} n_z + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{w}}{\partial z} n_y - \tilde{p} n_y \right] \right\} d\Gamma_e \qquad (4.21\text{-}38)$$

$$C^3{}_j = \int_{\Gamma_e} \left\{ N_i \frac{(1-\alpha)\Delta t}{\rho} \left[\frac{1}{2}\Delta t c^2 \frac{\partial \tilde{u}}{\partial x} n_z + \eta \frac{\partial \tilde{u}}{\partial z} n_x + \frac{1}{2}\Delta t c^2 \frac{\partial \tilde{v}}{\partial y} n_z + \eta \frac{\partial \tilde{v}}{\partial z} n_y - \tilde{p} n_z \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial x} n_x + \eta \frac{\partial \tilde{w}}{\partial y} n_y + \left(2\eta + \frac{1}{2}\Delta t c^2 \right) \frac{\partial \tilde{w}}{\partial z} n_z \right] \right\} d\Gamma_e \qquad (4.21\text{-}39)$$

$$C^4{}_j = \int_{\Gamma_e} \left\{ N_i \rho (1-\alpha) \Delta t^2 \frac{1}{2} c^2 \left(\frac{\partial \tilde{p}}{\partial x} n_x + \frac{\partial \tilde{p}}{\partial y} n_y + \frac{\partial \tilde{p}}{\partial z} n_z \right) \right\} d\Gamma_e \qquad (4.21\text{-}40)$$

The described scheme can also be used to solve steady-state conditions through an iterative algorithm. The U-V-W-P formulation is favoured by some researchers as the most straightforward finite element procedure for solution of the non-linear

Navier-Stokes and Stokes equations. Among the advantages list by Huebner *et al.* (2001) are as follows:

(1) only C^0 is required of the element interpolation functions, pressure

(2) pressure, velocity, velocity gradient and stress boundary condition can be directly incorporated into the matrix equations

(3) free surface problems are tractable

To provide a safeguard for the simulations in this study, a second scheme is employed in this study. The scheme and its formulation will be the subject of discussion in the section that follows.

4.5.2 Continuous Penalty Method

Another scheme commonly used to solve the equations of continuity and motion is the continuous Penalty Method (Hughes *et al.*, 1979; Bercovier and Engelman, 1979; Reddy, 1982; Reddy, 1982). In this approach, the penalty function formulation eliminates the pressure as an unknown variable through the use of a 'penalty' parameter and solves modified momentum equations for the velocity components. The pressure is represented by (Huebner *et al.*, 2001; Nassehi, 2002)

$$p = -\lambda\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z}\right) \tag{4.22}$$

where λ is the penalty parameter. In a viscous flow, if the parameter λ is specified to have a large numerical value in the solution, the flow incompressibility condition will be approximately satisfied as Equation (4.22) represents a perturbed form of the continuity equation. The principal advantage of the penalty formulation is that the additional flow variable p is eliminated and so is the need for solving the continuity equation. Depending whether the described substitution of pressure is carried out before or after the discretization of the governing equations, two different types of the penalty method are developed. In the continuous penalty method, the pressure term is substituted prior to the discretization of the governing equations. In the discrete penalty method, separate discretization for the equation of motion and the penalty equation are first obtained and then the pressure in the equation of motion is substituted using these discretized forms (Nassehi, 2002). The continuous penalty method is adapted in this study. In general, this scheme yields an equation that is overwhelmed by its penalty terms. To overcome this difficulty, the penalty sub-matrix in the elemental coefficient matrix is forced to become singular. This is achieved by adapting a 'reduced integration' to calculate the penalty terms in the elemental coefficient matrix (Zienkiewicz and Taylor, 2000).

Incorporating Equation (4.22) into Equations (3.8a), (3.8b), (3.8c) gives

$$\rho\frac{\partial v_x}{\partial t}=\lambda\frac{\partial}{\partial x}\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)+\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z} \qquad (4.23a)$$

$$\rho\frac{\partial v_y}{\partial t}=\lambda\frac{\partial}{\partial y}\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)+\frac{\partial \tau_{yx}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z} \qquad (4.23b)$$

$$\rho\frac{\partial v_z}{\partial t}=\lambda\frac{\partial}{\partial z}\left(\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}\right)+\frac{\partial \tau_{zx}}{\partial x}+\frac{\partial \tau_{zy}}{\partial y}+\frac{\partial \tau_{zz}}{\partial z} \qquad (4.23c)$$

Approximation of the unknown variables v_x, v_y, and v_z over an element has been expressed earlier in Equations (4.16a, b, c) as

$$v_x=u\approx\tilde{u}=\sum_{j=1}^{n}N_j u_j\ ,\ v_y=v\approx\tilde{v}=\sum_{j=1}^{n}N_j v_j\ ,\ v_z=w\approx\tilde{w}=\sum_{j=1}^{n}N_j w_j$$

The residual obtained after the differential equation is weighted and integrated over each element can be expressed by

$$\begin{aligned}\int_{\Omega_e} N_i \Bigg\{ & \rho \frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial t} - \lambda \frac{\partial}{\partial x}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z} \right) - \frac{\partial}{\partial x}\left[2\eta \frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} \right] \\ & - \frac{\partial}{\partial y}\left[\eta \left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial x} \right) \right] - \frac{\partial}{\partial z}\left[\eta \left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial z} \right) \right] \Bigg\} d\Omega_e = 0\end{aligned}$$

(4.24a)

$$\begin{aligned}\int_{\Omega_e} N_i \Bigg\{ & \rho \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial t} - \lambda \frac{\partial}{\partial y}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z} \right) - \frac{\partial}{\partial x}\left[\eta \left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial x} \right) \right] \\ & - \frac{\partial}{\partial y}\left[2\eta \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} \right] - \frac{\partial}{\partial z}\left[\eta \left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial z} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial y} \right) \right] \Bigg\} d\Omega_e = 0\end{aligned}$$

(4.24b)

$$\begin{aligned}\int_{\Omega_e} N_i \Bigg\{ & \rho \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial t} - \lambda \frac{\partial}{\partial z}\left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z} \right) - \frac{\partial}{\partial x}\left[\eta \left(\frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial z} \right) \right] \\ & - \frac{\partial}{\partial y}\left[\eta \left(\frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial z} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial y} \right) \right] - \frac{\partial}{\partial z}\left[2\eta \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z} \right] \Bigg\} d\Omega_e = 0\end{aligned}$$

(4.24c)

Integration by parts using Gauss theorem to the second order derivatives in Equations (4.24a), (4.24b), (4.24c) yield

$$\begin{aligned}
&\int_{\Omega_e}\left\{\rho N_i\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial t}+\lambda\frac{\partial N_i}{\partial x}\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right)+\frac{\partial N_i}{\partial x}\left(2\eta\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}\right)\right.\\
&\left.+\frac{\partial N_i}{\partial y}\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial x}\right)\right]+\frac{\partial N_i}{\partial z}\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial z}\right)\right]\right\}d\Omega_e\\
&+\int_{\Gamma_e}N_i\left\{-\lambda\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right)\hat{n}_x-\left(2\eta\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}\right)\hat{n}_x\right.\\
&\left.-\eta\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial x}\right)\hat{n}_y-\eta\left(\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial z}\right)\hat{n}_z\right\}d\Gamma_e=0
\end{aligned}$$

(4.25a)

$$
\begin{aligned}
&\int_{\Omega_e}\left\{N_i\rho\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial t}+\lambda\frac{\partial N_i}{\partial y}\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right)+\frac{\partial N_i}{\partial x}\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial x}\right)\right]\right.\\
&\qquad\left.+\frac{\partial N_i}{\partial y}\left[2\eta\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}\right]+\frac{\partial N_i}{\partial z}\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial z}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial y}\right)\right]\right\}d\Omega_e\\
&+\int_{\Gamma_e}N_i\left\{-\lambda\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right)\hat{n}_y-\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial x}\right)\hat{n}_x\right]\right.\\
&\qquad\left.-\left[2\eta\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}\right]\hat{n}_y-\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial z}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial y}\right)\hat{n}_z\right]\right\}d\Gamma_e=0
\end{aligned}
$$

(4.25b)

$$
\begin{aligned}
&\int_{\Omega_e}\Bigg\{\rho N_i\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial t}+\lambda\frac{\partial N_i}{\partial z}\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right)+\frac{\partial N_i}{\partial x}\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial z}\right)\right]\\
&\qquad+\frac{\partial N_i}{\partial y}\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial z}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial y}\right)\right]+\frac{\partial N_i}{\partial z}\left[2\eta\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right]\Bigg\}d\Omega_e\\
&\qquad+\int_{\Gamma_e}N_i\Bigg\{-\lambda\left(\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial y}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right)\hat{n}_z-\Bigg[\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial x}+\frac{\partial\sum_{j=1}^{n}N_ju_j}{\partial z}\right)\hat{n}_x\right]\\
&\qquad-\left[\eta\left(\frac{\partial\sum_{j=1}^{n}N_jv_j}{\partial z}+\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial y}\right)\hat{n}_y\right]-\left[2\eta\frac{\partial\sum_{j=1}^{n}N_jw_j}{\partial z}\right]\hat{n}_z\Bigg]\Bigg\}d\Gamma_e=0
\end{aligned}
$$

(4.25c)

In order to preserve the continuity in non-Newtonian fluid flow problems, it is necessary to maintain a balance between the viscosity and the penalty parameter. The penalty parameter can be related to the viscosity as (Nakazawa *et al.*, 1982)

$$\lambda=\lambda_0\eta \qquad (4.26)$$

where λ_0 is a large dimensionless parameter and η is the local viscosity. The typical recommended value for λ_0 is approximately 10^8 (Nassehi, 2002). The penalty method has been considered a cost effective scheme as the approach is easier to be programmed and less computing storage is required. With the pressure eliminated as a field unknown, there are also fewer equations to solve.

4.5.2.1 Calculation of Pressure

After the velocity components have been obtained, the pressure can be calculated using Equation (4.22) through a variational recovery method. The discretized form of Equation (4.22) in conjunction with the Galerkin finite element discretization is given as

$$\int_{\Omega_e} N_i \sum_{j=1}^{n} N_j p_j d\Omega_e = -\int_{\Omega_e} N_i \lambda \left(\frac{\partial \sum_{j=1}^{n} N_j u_j}{\partial x} + \frac{\partial \sum_{j=1}^{n} N_j v_j}{\partial y} + \frac{\partial \sum_{j=1}^{n} N_j w_j}{\partial z} \right) d\Omega_e \tag{4.27}$$

The coefficient matrix on the left hand side of the Equation (4.23) is the mass matrix

$$M_{ij} = \int_{\Omega_e} N_i N_j d\Omega_e \tag{4.28}$$

This matrix is diagonalized using a simple mass lumping technique (Pittman and Nakazawa, 1984) to reduce the computer resources required for the solution for pressure in this method. There are various schemes for lumping and Huebner *et al.* (2001) gave a practical comparison of these lumping methods.

4.5.2.2 Solution of the Transient Equations

As discussed earlier, the inclusion of transient terms forces the use of a time stepping scheme in the solution of the flow equations. The *implicit θ time-stepping technique* was used in conjunction with the Penalty Method in this study. In this technique, the time derivatives in the differential equations are kept unchanged and the spatial discretization is carried out to form a weighted residual statement in the usual manner, as has been shown in the earlier section. As a result, after the spatial discretization, instead of a set of algebraic equations which are normally derived for steady-state problems, a system of equations with time derivatives are generated. There are two families of algorithm associated with this technique, namely implicit and explicit algorithms. Although the implicit family of algorithms requires considerably more computational effort for transient solution than for the corresponding equilibrium problem, it is nevertheless more advantages than the explicit algorithm. In an explicit algorithm, although the nodal unknowns at each time are computed from uncoupled algebraic equations thus requiring substantially less computational effort, the time step Δt must be selected to be less than a critical value for the response to remain stable. If the time step for the explicit algorithm is selected arbitrarily, the computed response may become unstable, and the computed values will grow without bound as time increases (Huebner *et al.*, 2001). The implicit algorithm has been used in the present study.

Taking t_θ to be the typical time between t_n and t_{n+1} so that $t_\theta = t_n + t_{n+1}$, the system is given as

$$[M]\{\dot{X}\}_\theta + [K]\{X\}_\theta = [\{F\}]_\theta \tag{4.29}$$

where subscript θ indicates that the weighted residual statement is derived at time level $0 \leq \theta \leq 1$, as shown in Figure 4.2.

The temporal derivative term in Equation (4.29) is approximated by a forward difference as

$$\{\dot{X}\} = \frac{\{X\}_{n+1} - \{X\}_n}{\Delta t} \tag{4.30}$$

The remaining terms in Equation (4.29) are approximated using a linear interpolation as

$$\{X\}_\theta = (1-\theta)\{X\}_n + \theta\{X\}_{n+1} \tag{4.31a}$$

$$\{F\}_\theta = (1-\theta)\{F\}_n + \theta\{F\}_{n+1} \tag{4.32b}$$

Substitution from Equations (4.30), (4.31a) and (4.32b) into Equation (4.29) and carrying out algebraic manipulations gives

$$\left\{\left[M_{ij}\right]+\theta\Delta t\left[K_{ij}\right]_{n+1}\right\}\left\{X\right\}_{n+1}-\left\{\left[M_{ij}\right]-(1-\theta)\Delta t\left[K_{ij}\right]_{n}\right\}\left\{X\right\}_{n}$$

$$=\left[(1-\theta)\left\{F_j\right\}_n+\theta\left\{F_j\right\}_{n+1}\right]\Delta t \qquad (4.32)$$

If θ = 0, the algorithm is the forward difference method (Euler method); if θ = 0.5, the algorithm is Crank-Nicolson method or the central difference method; if θ = 2/3, the algorithm is the Galerkin method; if θ = 1.0, the algorithm is backward difference method, or the explicit method. The choice of the best value of θ depends on the rate of convergence, accuracy and the stability desired.

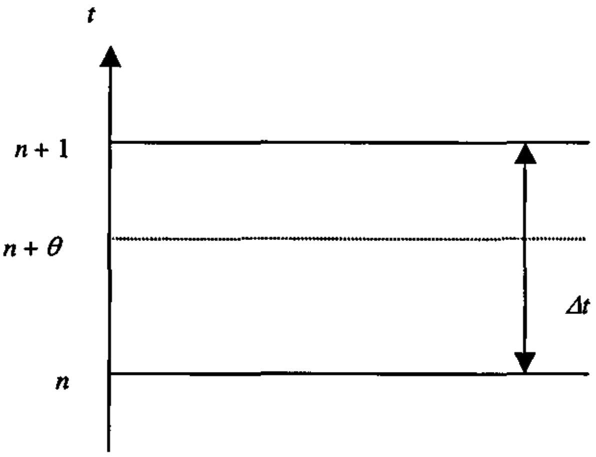


Figure 4.2: Time stepping scheme.

The final working equations for the Penalty Method can be written as

$$\begin{bmatrix} L^{11}{}_{ij} & L^{12}{}_{ij} & L^{13}{}_{ij} \\ L^{21}{}_{ij} & L^{22}{}_{ij} & L^{23}{}_{ij} \\ L^{31}{}_{ij} & L^{32}{}_{ij} & L^{33}{}_{ij} \end{bmatrix}^{n+1} \begin{Bmatrix} u_j \\ v_j \\ w_j \end{Bmatrix}^{n+1} =$$

$$\begin{bmatrix} Q^{11}{}_{ij} & Q^{12}{}_{ij} & Q^{13}{}_{ij} \\ Q^{21}{}_{ij} & Q^{22}{}_{ij} & Q^{23}{}_{ij} \\ Q^{31}{}_{ij} & Q^{32}{}_{ij} & Q^{33}{}_{ij} \end{bmatrix}^{n} \begin{Bmatrix} u_j \\ v_j \\ w_j \end{Bmatrix}^{n} + \begin{Bmatrix} D^{1}{}_{j} \\ D^{2}{}_{j} \\ D^{3}{}_{j} \end{Bmatrix}^{n+1} + \begin{Bmatrix} E^{1}{}_{j} \\ E^{2}{}_{j} \\ E^{3}{}_{j} \end{Bmatrix}^{n} \qquad (4.33)$$

where

$$L^{11}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{\theta\Delta t}{\rho}\left[(2\eta+\lambda)\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\} dxdydz \tag{4.34-1}$$

$$L^{12}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{\theta\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x} + \lambda\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}\right)\right\} dxdydz \tag{4.34-2}$$

$$L^{13}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{\theta\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x} + \lambda\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z}\right)\right\} dxdydz \tag{4.34-3}$$

$$L^{21}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{\theta\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y} + \lambda\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x}\right)\right\} dxdydz \tag{4.34-4}$$

$$L^{22}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{\theta\Delta t}{\rho}\left[\eta\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + (2\eta+\lambda)\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\} dxdydz \tag{4.34-5}$$

$$L^{23}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{\theta\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial y} + \lambda\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial z}\right)\right\} dxdydz \tag{4.34-6}$$

$$L^{31}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{\theta\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial z} + \lambda\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x}\right)\right\} dxdydz \tag{4.34-7}$$

$$L^{32}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{\theta\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial z} + \lambda\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial y}\right)\right\} dxdydz \tag{4.34-8}$$

$$L^{33}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{\theta\Delta t}{\rho}\left[\eta\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + (2\eta+\lambda)\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\} dxdydz \tag{4.34-9}$$

$$Q^{11}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{(1-\theta)\Delta t}{\rho}\left[(2\eta+\lambda)\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\} dxdydz \tag{4.34-10}$$

$$Q^{12}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{(1-\theta)\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial x} + \lambda\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial y}\right)\right\} dxdydz \tag{4.34-11}$$

$$Q^{13}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{(1-\theta)\Delta t}{\rho}\left(\eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial x} + \lambda\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}\right)\right\} dxdydz \tag{4.34-12}$$

$$Q^{22}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{(1-\theta)\Delta t}{\rho}\left[\eta\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + (2\eta+\lambda)\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \eta\frac{\partial N_i}{\partial z}\frac{\partial N_j}{\partial z}\right]\right\} dxdydz$$

$$\tag{4.34-13}$$

$$Q^{23}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{(1-\theta)\Delta t}{\rho} \left(\eta \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial y} + \lambda \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial z} \right) \right\} dxdydz \tag{4.34-14}$$

$$Q^{31}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{(1-\theta)\Delta t}{\rho} \left(\eta \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial z} + \lambda \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial x} \right) \right\} dxdydz \tag{4.34-15}$$

$$Q^{32}{}_{ij} = \iiint_{\Omega_e} \left\{ \frac{(1-\theta)\Delta t}{\rho} \left(\eta \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial z} + \lambda \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial y} \right) \right\} dxdydz \tag{4.34-16}$$

$$Q^{33}{}_{ij} = \iiint_{\Omega_e} \left\{ N_i N_j + \frac{(1-\theta)\Delta t}{\rho} \left(\eta \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \eta \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + (2\eta + \lambda) \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z} \right) \right\} dxdydz \tag{4.34-17}$$

$$D^1{}_j = \int_{\Gamma_e} \left\{ N_i \frac{\theta \Delta t}{\rho} \left[(2\eta + \lambda) \frac{\partial \tilde{u}}{\partial x} n_x + \eta \frac{\partial \tilde{u}}{\partial y} n_y + \eta \frac{\partial \tilde{u}}{\partial z} n_z + \eta \frac{\partial \tilde{v}}{\partial x} n_y + \lambda \frac{\partial \tilde{v}}{\partial y} n_z \right.\right.$$
$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial x} n_z + \lambda \frac{\partial \tilde{w}}{\partial z} n_x \right] \right\} d\Gamma \tag{4.34-18}$$

$$D^2{}_j = \int_{\Gamma_e} \left\{ N_i \frac{\theta \Delta t}{\rho} \left[\lambda \frac{\partial \tilde{u}}{\partial x} n_y + \eta \frac{\partial \tilde{u}}{\partial y} n_x + \eta \frac{\partial \tilde{v}}{\partial x} n_x + (2\eta + \lambda) \frac{\partial \tilde{v}}{\partial y} n_y + \eta \frac{\partial \tilde{v}}{\partial z} n_z \right.\right.$$
$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial y} n_z + \lambda \frac{\partial \tilde{w}}{\partial z} n_y \right] \right\} d\Gamma \tag{4.34-19}$$

$$D^3{}_j = \int_{\Gamma_e} \left\{ N_i \frac{\theta \Delta t}{\rho} \left[\lambda \frac{\partial \tilde{u}}{\partial x} dn_z + \eta \frac{\partial \tilde{u}}{\partial z} dn_x + \lambda \frac{\partial \tilde{v}}{\partial y} dn_z + \eta \frac{\partial \tilde{v}}{\partial z} dn_y + \eta \frac{\partial \tilde{w}}{\partial x} n_x \right.\right.$$
$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial y} n_y + (2\eta + \lambda) \frac{\partial \tilde{w}}{\partial z} n_z \right] \right\} d\Gamma \tag{4.34-20}$$

$$E^1{}_j = \int_{\Gamma_e} \left\{ N_i \frac{(1-\theta)\Delta t}{\rho} \left[(2\eta + \lambda) \frac{\partial \tilde{u}}{\partial x} n_x + \eta \frac{\partial \tilde{u}}{\partial y} n_y + \eta \frac{\partial \tilde{u}}{\partial z} n_z + \eta \frac{\partial \tilde{v}}{\partial x} n_y + \lambda \frac{\partial \tilde{v}}{\partial y} n_z \right.\right.$$
$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial x} n_z + \lambda \frac{\partial \tilde{w}}{\partial z} n_x \right] \right\} d\Gamma \tag{4.34-21}$$

$$E^2{}_j = \int_{\Gamma_e} \left\{ N_i \frac{(1-\theta)\Delta t}{\rho} \left[\lambda \frac{\partial \tilde{u}}{\partial x} n_y + \eta \frac{\partial \tilde{u}}{\partial y} n_x + \eta \frac{\partial \tilde{v}}{\partial x} n_x + (2\eta + \lambda) \frac{\partial \tilde{v}}{\partial y} n_y + \eta \frac{\partial \tilde{v}}{\partial z} n_z \right.\right.$$
$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial y} n_z + \lambda \frac{\partial \tilde{w}}{\partial z} n_y \right] \right\} d\Gamma \tag{4.34-22}$$

$$E^3{}_j = \int_{\Gamma_e} \left\{ N_i \frac{(1-\theta)\Delta t}{\rho} \left[\lambda \frac{\partial \tilde{u}}{\partial x} dn_z + \eta \frac{\partial \tilde{u}}{\partial z} dn_x + \lambda \frac{\partial \tilde{v}}{\partial y} dn_z + \eta \frac{\partial \tilde{v}}{\partial z} dn_y + \eta \frac{\partial \tilde{w}}{\partial x} n_x \right.\right.$$

$$\left.\left. + \eta \frac{\partial \tilde{w}}{\partial y} n_y + (2\eta + \lambda) \frac{\partial \tilde{w}}{\partial z} n_z \right] \right\} d\Gamma \qquad (4.34\text{-}23)$$

4.6 Convergence of the Solution

Since numerical modelling is iterative, it is important to know when to terminate the iterative loop. One convergence criterion is to monitor absolute and relative changes of the objective and constraint functions and the design parameters (Vanderplaats, 1999). Convergence may then be indicated when changes in the performance measures and/or design parameters between successive iterations are within a predefined tolerance. In this study, convergence is checked using a calculated ratio of the Euclidean norm (Lapidus and Pinder, 1982) between successive iterations to the norm of the solution via the following equation:

$$\sqrt{\frac{\sum_{i=1}^{N} \left| X_i^{r+1} - X_i^{r} \right|^2}{\sum_{i=1}^{N} \left| X_i^{r+1} \right|^2}} \le E \qquad (4.35)$$

where X is the field unknown, r is the number of iteration cycle, N is the total number of degrees of freedom and E is predefined convergence tolerance.

It is necessary to note that all numerical computations involve round-off errors. This error increases as the number of calculations in the solution procedure is increased. Therefore, successive mesh refinement that increase the number of finite element calculations do not necessarily lead to more accurate solutions. However, one may assume a theoretical situation where the rounding error is eliminated (Nassehi, 2002).

4.7 Mesh Refinement

The standard procedure for improving the accuracy of a finite element approximation is to refine the computational grid. This is achieved by using a smaller element size to pack a larger number of meshes in the same domain area or volume. This also provides a practical method for testing the convergence in the solution of non-linear problems through the comparison of the results in successive mesh refinements. There are several adaptive refinement methods and they include the *h*- method, the *p*- method, the *r*- method and the combined *h*/*p* method. In the *h*-method (Demkowicz *et al.*, 1985; Löhner et al., 1985; Löhner, 1987; Ramakrishnan *et al.*, 1990; Ramakrishnan *et al.*, 1992), the elements of the initial mesh are refined into smaller elements or de-refined into larger elements. The number and the size of elements vary with each level of refinement. In the *p*- method (Zienkiewicz *et al.*, 1983; Peano, 1976; Peano *et al.*, 1979; Szabó, 1979; Szabó, 1986), the order of the polynomial used for the element interpolation function is increased or decreased while keeping the number and geometry of the element constant. The *r*- method (Oden *et al.*, 1986) keeps the number of elements and their connenctivity constant but moves the nodal position. There are also methods of mesh refinement that use combinations of the three methods mentioned. For example, the *h*/*p* method (Demkowicz *et al.*, 1989; Oden *et al.*, 1989; Rachowicz *et al.*, 1989; Tworzydlo *et al.*, 1992) that refined or de-refined some elements while increasing or decreasing the order of interpolation polynomials in other elements. A detailed account of comparison of these mesh refinement methods can be found at Huebner *et al.* (2001).

In this study, *h*- method of mesh refinement has been employed to improve the accuracy of the finite element solution. It should be noted that it is no trivial matter to increase the mesh density for a three-dimensional domain where it is normal for the number of elements to increase an order of magnitude at each level of mesh refinement.

4.8 Output

The availability of commercial post-processing graphic tools to view the pressure contours and velocity vectors has enabled the analysis to be conducted with great convenience. Raw simulation data generated from the developed algorithm were analyzed and checked both numerically and graphically to verify the results. In this study, the pressure contours and the velocity vectors of domain cross-sections were plotted using SURFER® 8 graphic package.

4.9 Developed Algorithm

Two computer codes based on the U-V-W-P scheme and the continuous Penalty scheme have been developed to solve the continuity and the Stokes equations in the three-dimensional domain. In the U-V-W-P technique, both velocity and pressure in the governing equations are regarded as primitive variables and are discretized as unknowns. In the continuous Penalty scheme, the primary variables are the velocity fields while the pressure is calculated through a variational recovery method. In both cases, the obtained results are then used to update the rheology properties of the fluid. The solution algorithm for the U-V-W-P and the continuous Penalty scheme are summarized in Figure 4.3 and Figure 4.4.

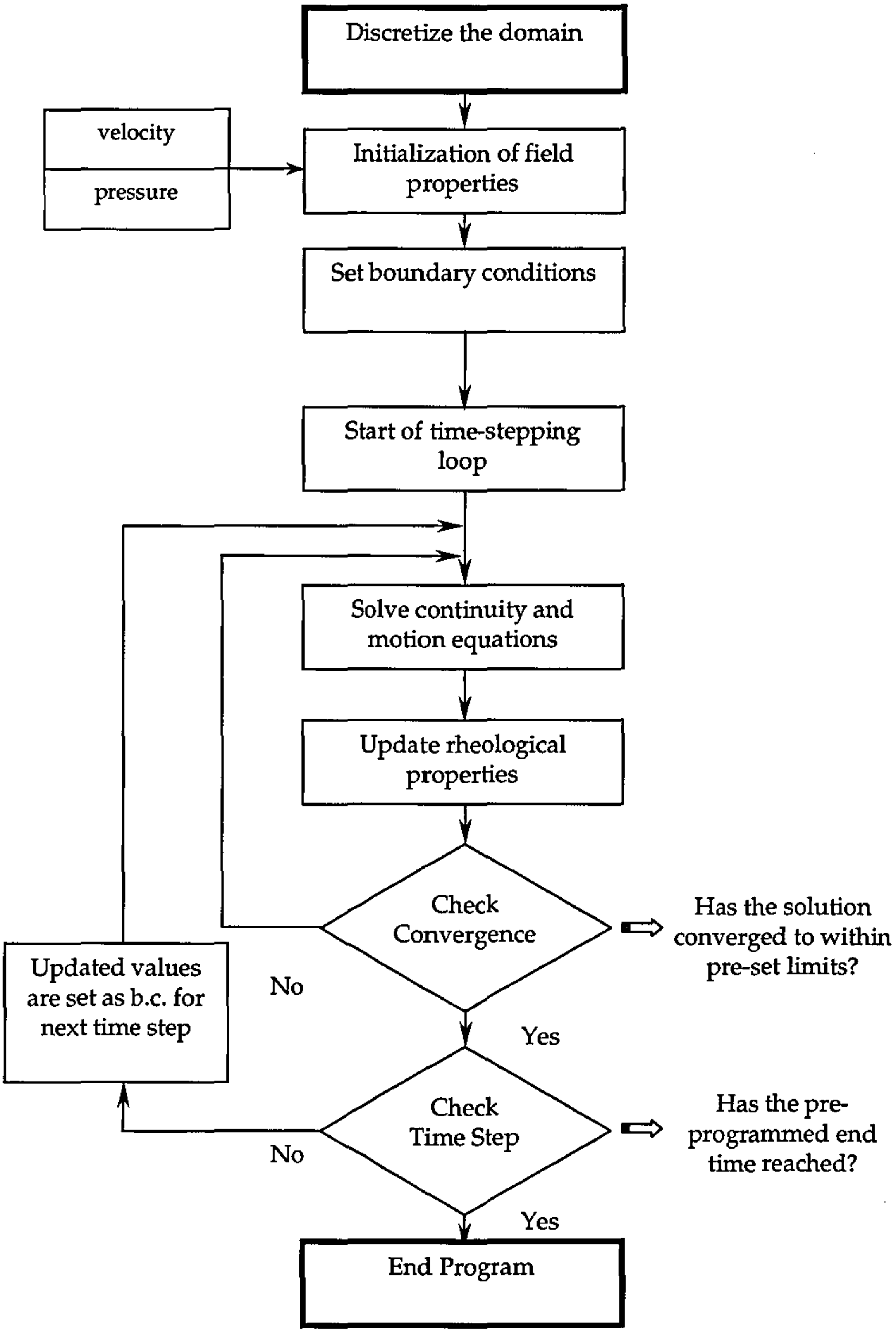

Discretize the domain
velocity
pressure
Initialization of field properties
Set boundary conditions
Start of time-stepping loop
Solve continuity and motion equations
Update rheological properties
Check Convergence
Has the solution converged to within pre-set limits?
No
Yes
Updated values are set as b.c. for next time step
Check Time Step
Has the pre-programmed end time reached?
No
Yes
End Program

Figure 4.3 The developed algorithm for U-V-W-P scheme.

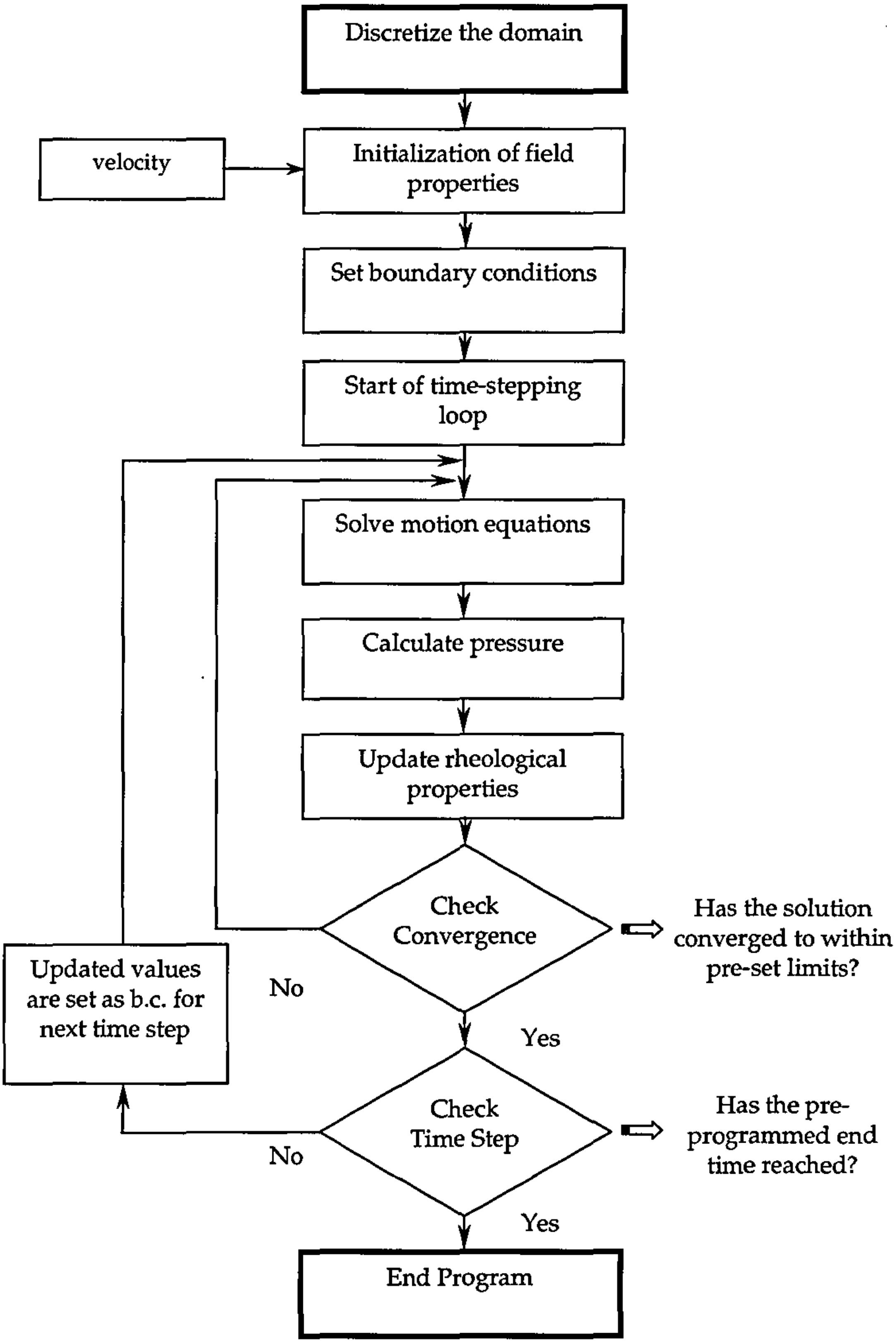

Discretize the domain
velocity
Initialization of field properties
Set boundary conditions
Start of time-stepping loop
Solve motion equations
Calculate pressure
Update rheological properties
Check Convergence
Has the solution converged to within pre-set limits?
No
Yes
Updated values are set as b.c. for next time step
Check Time Step
Has the pre-programmed end time reached?
No
Yes
End Program

Figure 4.4 The developed algorithm for continuous Penalty scheme.

4.10 Closure

In this chapter, the concept of the finite element method employed in this study was explained. The choice of interpolation and the restrictions of inf-sup conditions were briefly discussed as they directly affect the reliability of the model and the computing economy. The U-V-W-P scheme in conjunction with the Taylor Galerkin method and the continuous Penalty scheme for solving continuity and Stokes motion equation were detailed. The considerations for the derivation of the solution scheme in each case were presented. At the end of this chapter, the schematic diagram of the developed algorithm for each scheme was illustrated. In the next chapter, the domain selection for the three-dimensional wire mesh geometry, the numerical properties and the results from the simulations will be presented and discussed.

CHAPTER 5

RESULTS AND DISCUSSIONS

Computer codes based on FORTRAN-90 have been developed to execute the schemes described. The programs were compiled using Digital Visual Fortran version 6.0. The domain development and the finite element discretization of the mesh were handled using COSMOS™ GEOSTAR, a powerful pre-processing and post-processing software package developed by Structural Research and Analysis Corporation (SRAC), Los Angeles, USA. The COSMOS™ GEOSTAR output files that contain the finite element mesh data were processed using an in-house developed utility program FEUT20 to rearrange the data to a program-readable format. The physical properties and the boundary conditions for each domain were then included to become the INPUT files. The computations based on these INPUT files were then executed using a desktop with an Intel Pentium IV 2.6 GHz processor. The post-processing presentation of field variables such as flow velocity vectors and pressure contour were achieved by a commercial graphic software package SURFER® 8 developed by Golden Software, Colorado, USA.

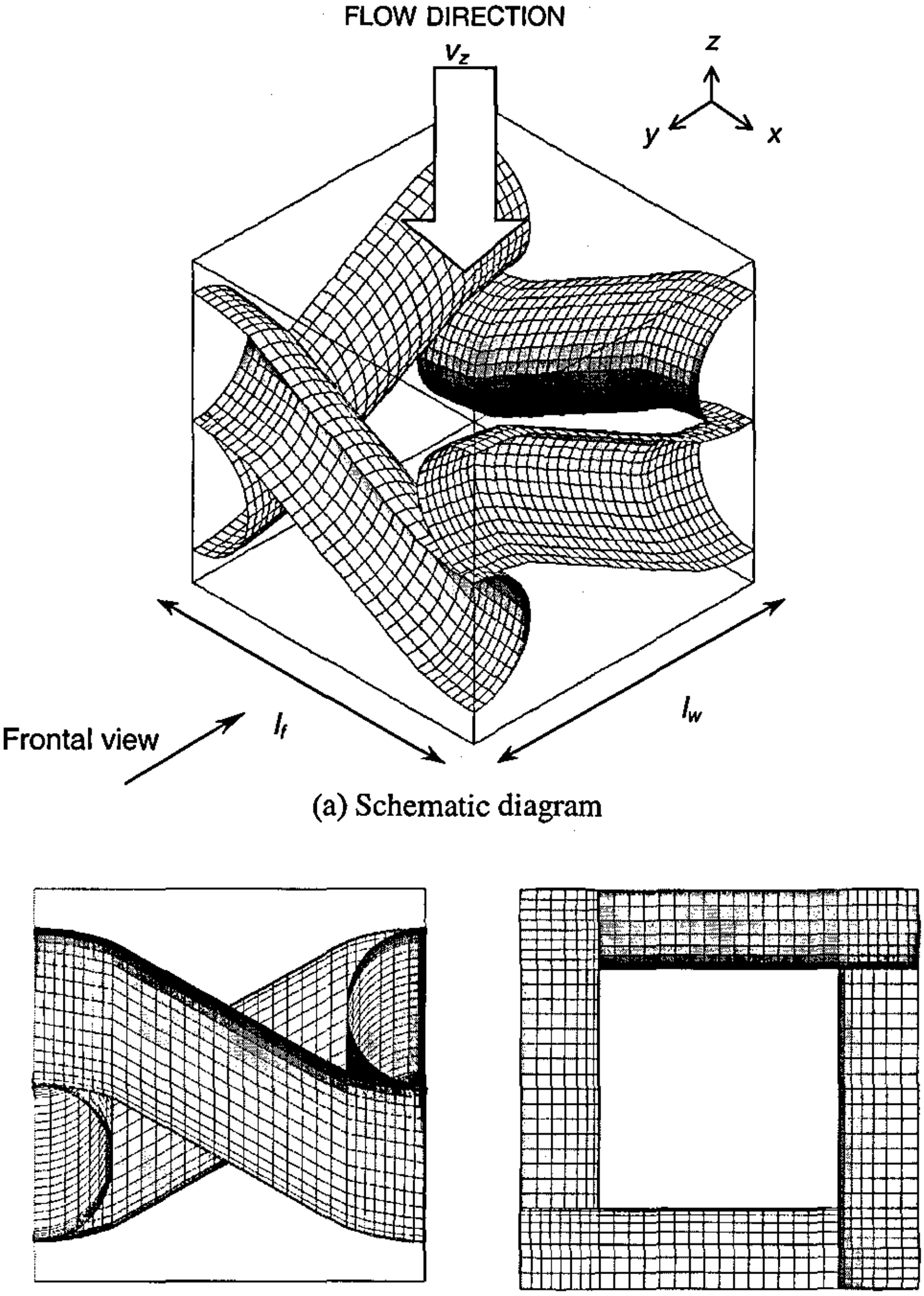
5.1 Domain Discretization and Boundary Conditions

5.1.1 Domain Studied and Boundary Conditions

A selection of domains based on the four basic pore types described earlier has been developed. A total of nine simulation domains have been designed and used to achieve the objective of this study. Descriptions for each of these domains will be given in this section.

(a) **DOMAIN 1**: Pore type 1 with aperture to diameter ratio 1.5.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:5:2. The mesh contains 4608 elements and 6560 nodes. The crimp diagrams for domain 1 are depicted in Figure 5.1.



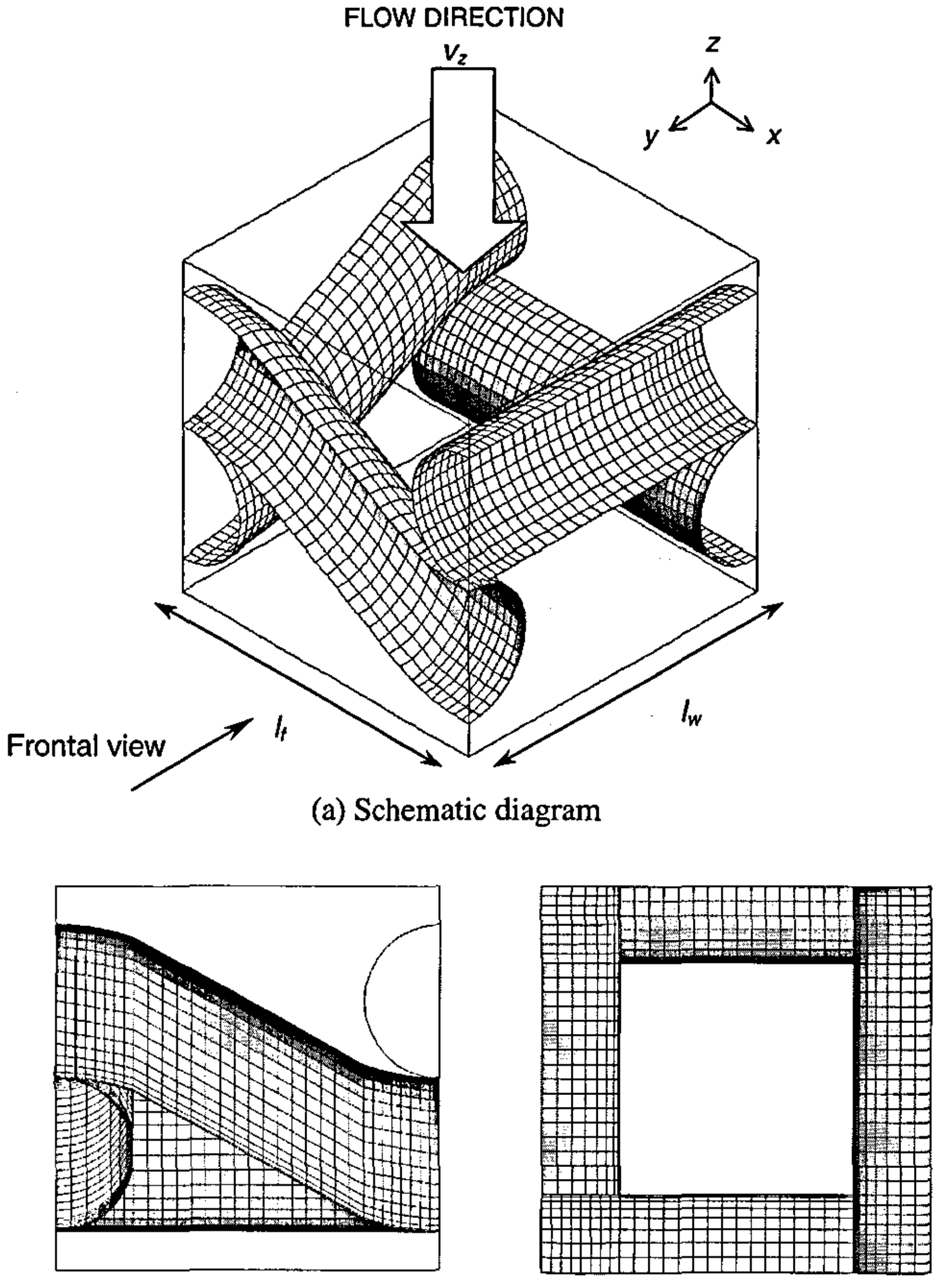
(a) Schematic diagram

(b) Frontal view

Figure 5.1 (a) Schematic diagram (b) frontal view and (c) top view of pore type 1 with aperture to diameter ratio 1.5.

(b) **DOMAIN 2**: Pore type 2 with aperture to diameter ratio 1.5.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:5:2. The mesh contains 4608 elements and 6514 nodes. The crimp diagrams for domain 2 are depicted in Figure 5.2.



(a) Schematic diagram

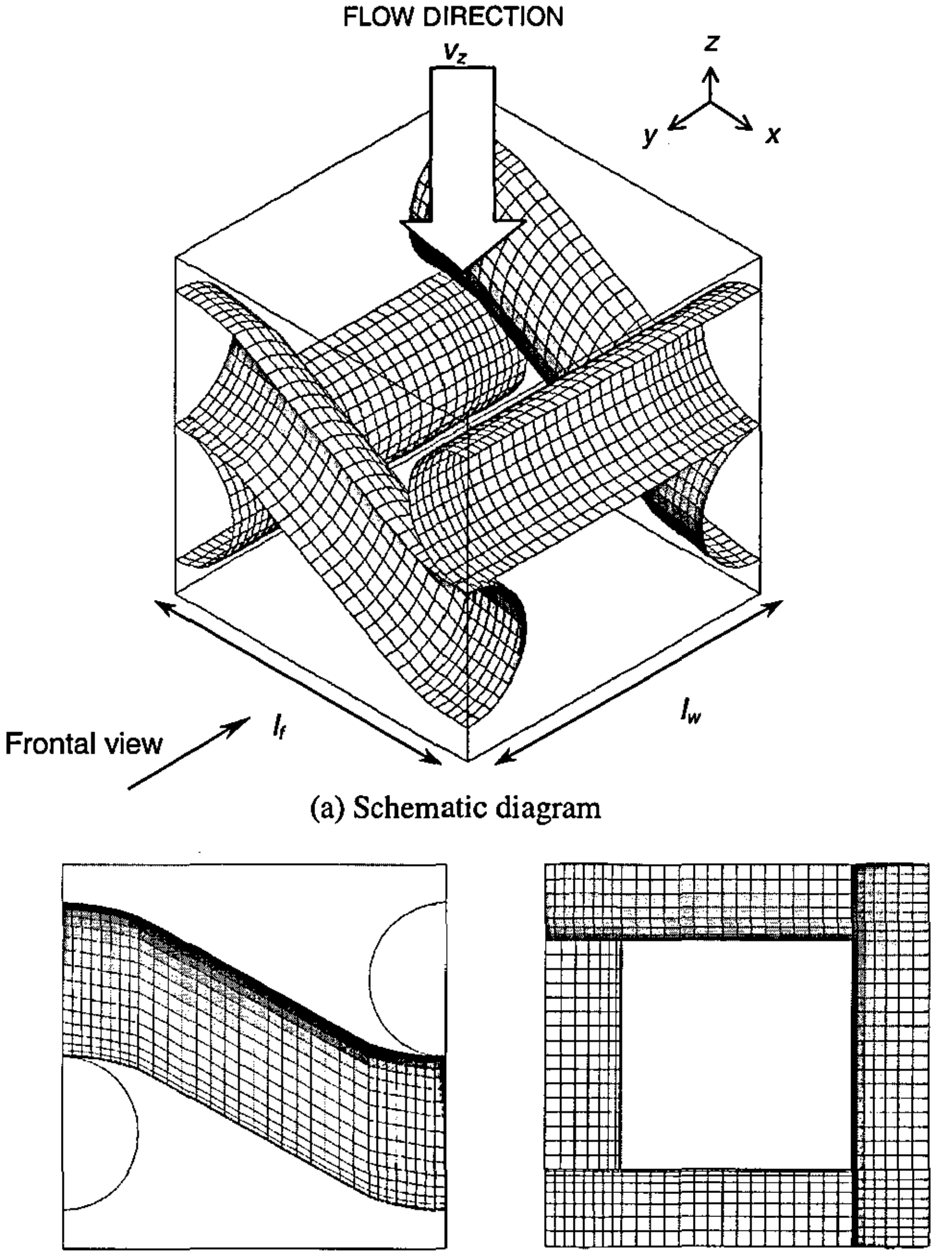
(b) Frontal view

(c) Top view

Figure 5.2 (a) Schematic diagram (b) frontal view and (c) top view of pore type 2 with aperture to diameter ratio 1.5.

(c) **DOMAIN 3**: Pore type 3 with aperture to diameter ratio 1.5.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:5:2. The mesh contains 4608 elements and 6527 nodes. The crimp diagrams for domain 3 are depicted in Figure 5.3.



(a) Schematic diagram

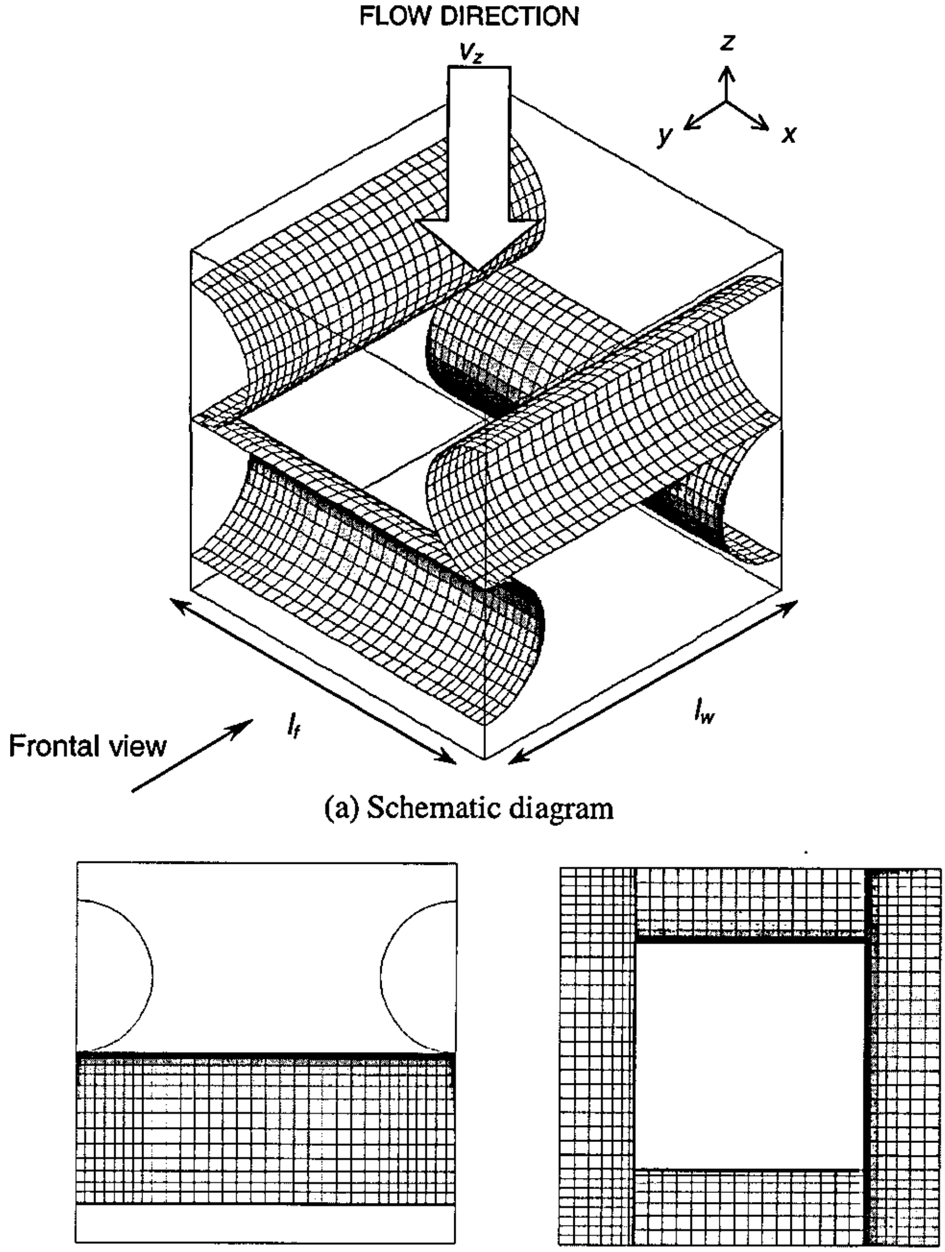
(b) Frontal view

(c) Top view

Figure 5.3 (a) Schematic diagram (b) frontal view and (c) top view of pore type 3 with aperture to diameter ratio 1.5.

(d) **DOMAIN 4:** Pore type 4 with aperture to diameter ratio 1.5.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:5:2. The mesh contains 4608 elements and 5357 nodes. The crimp diagrams for domain 4 are depicted in Figure 5.4.



(a) Schematic diagram

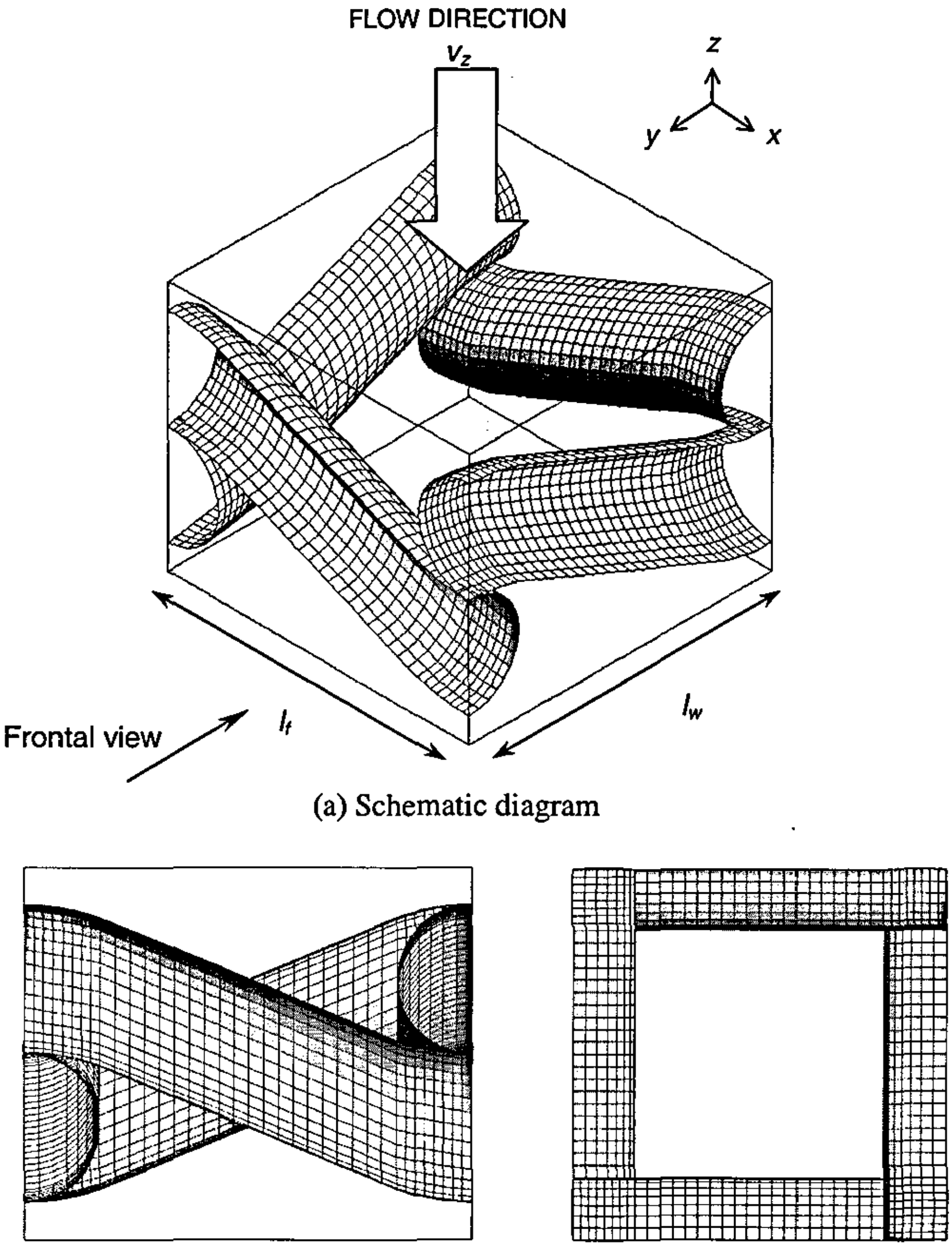
(b) Frontal view

(c) Top view

Figure 5.4 (a) Schematic diagram (b) frontal view and (c) top view of pore type 4 with aperture to diameter ratio 1.5.

(e) **DOMAIN 5**: Pore type 1 with aperture to diameter ratio 2.0.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 6:5:6:2. The mesh contains 5560 elements and 7778 nodes. The crimp diagrams for domain 5 are depicted in Figure 5.5.



(a) Schematic diagram

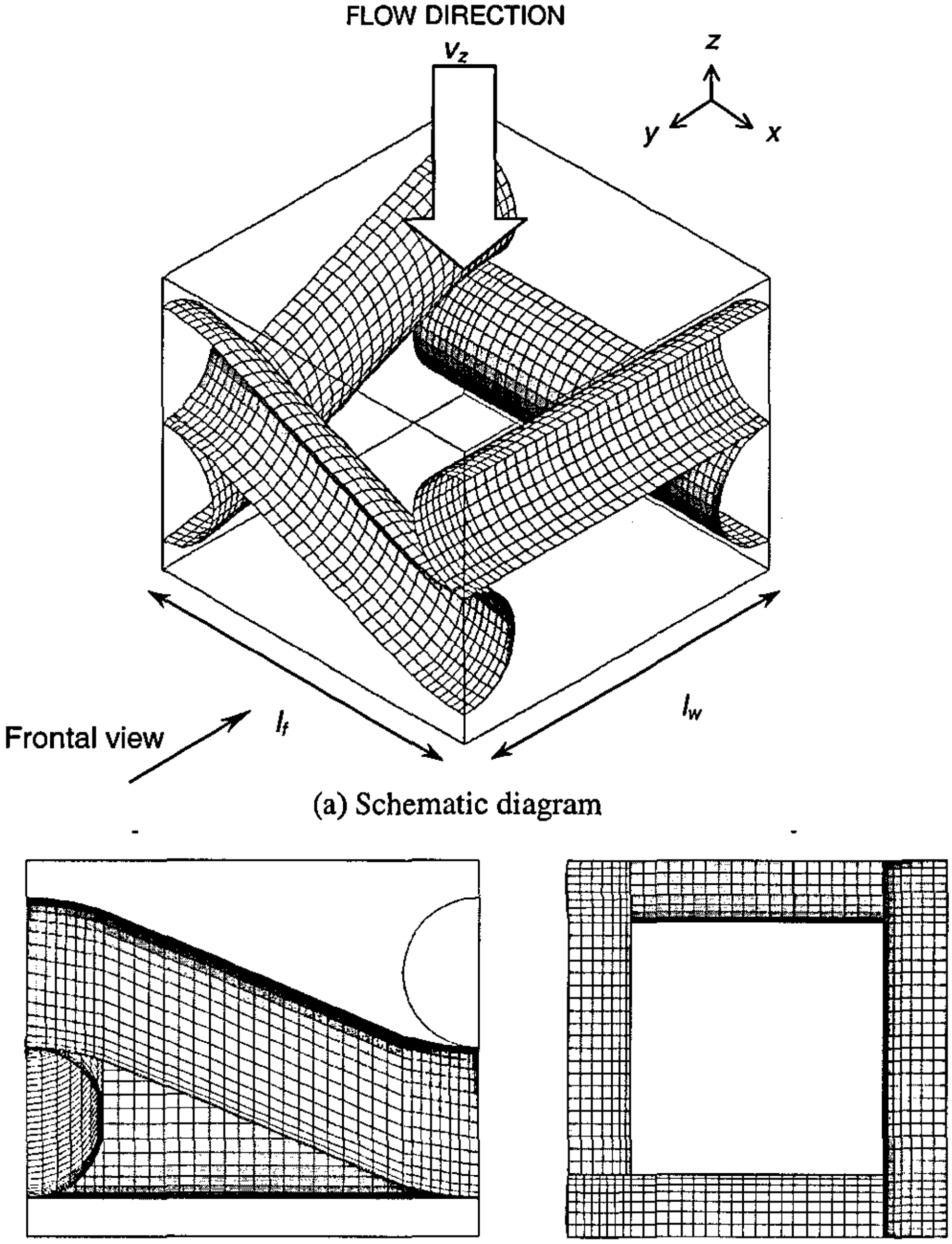
(b) Frontal view

(c) Top view

Figure 5.5 (a) Schematic diagram (b) frontal view and (c) top view of pore type 1 with aperture to diameter ratio 2.0.

(f) **DOMAIN 6**: Pore type 2 with aperture to diameter ratio 2.0.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 6:5:6:2. The mesh contains 5560 elements and 7722 nodes. The crimp diagrams for domain 6 are depicted in Figure 5.6.



(a) Schematic diagram

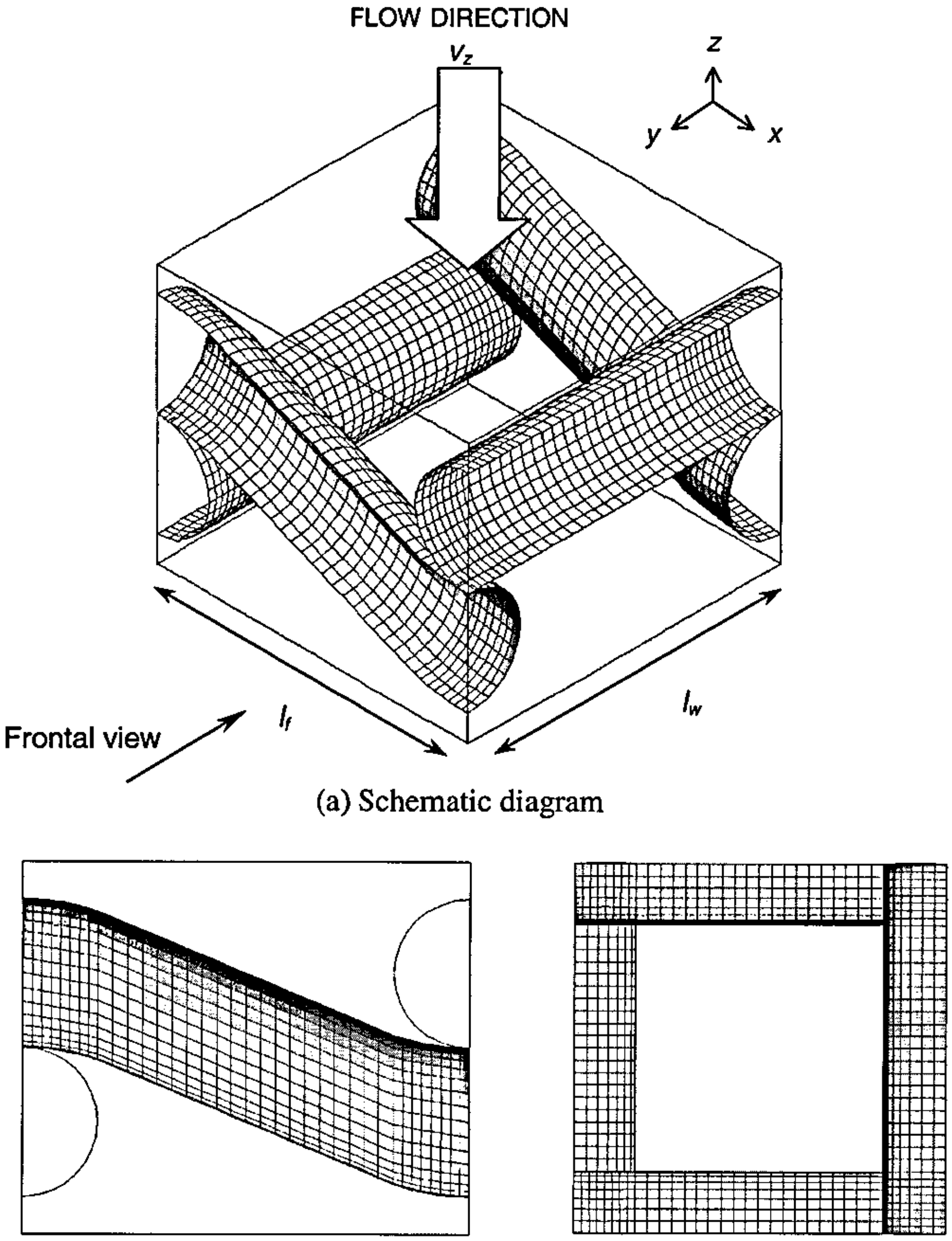
(b) Frontal view

(c) Top view

Figure 5.6 (a) Schematic diagram (b) frontal view and (c) top view of pore type 2 with aperture to diameter ratio 2.0.

(g) **DOMAIN 7**: Pore type 3 with aperture to diameter ratio 2.0.

The ratio of the domain's length (x), width (y), height (z) and wire diameter (d) is 6:5:6:2. The mesh contains 5560 elements and 7735 nodes. The crimp diagrams for domain 7 are depicted in Figure 5.7.



(a) Schematic diagram

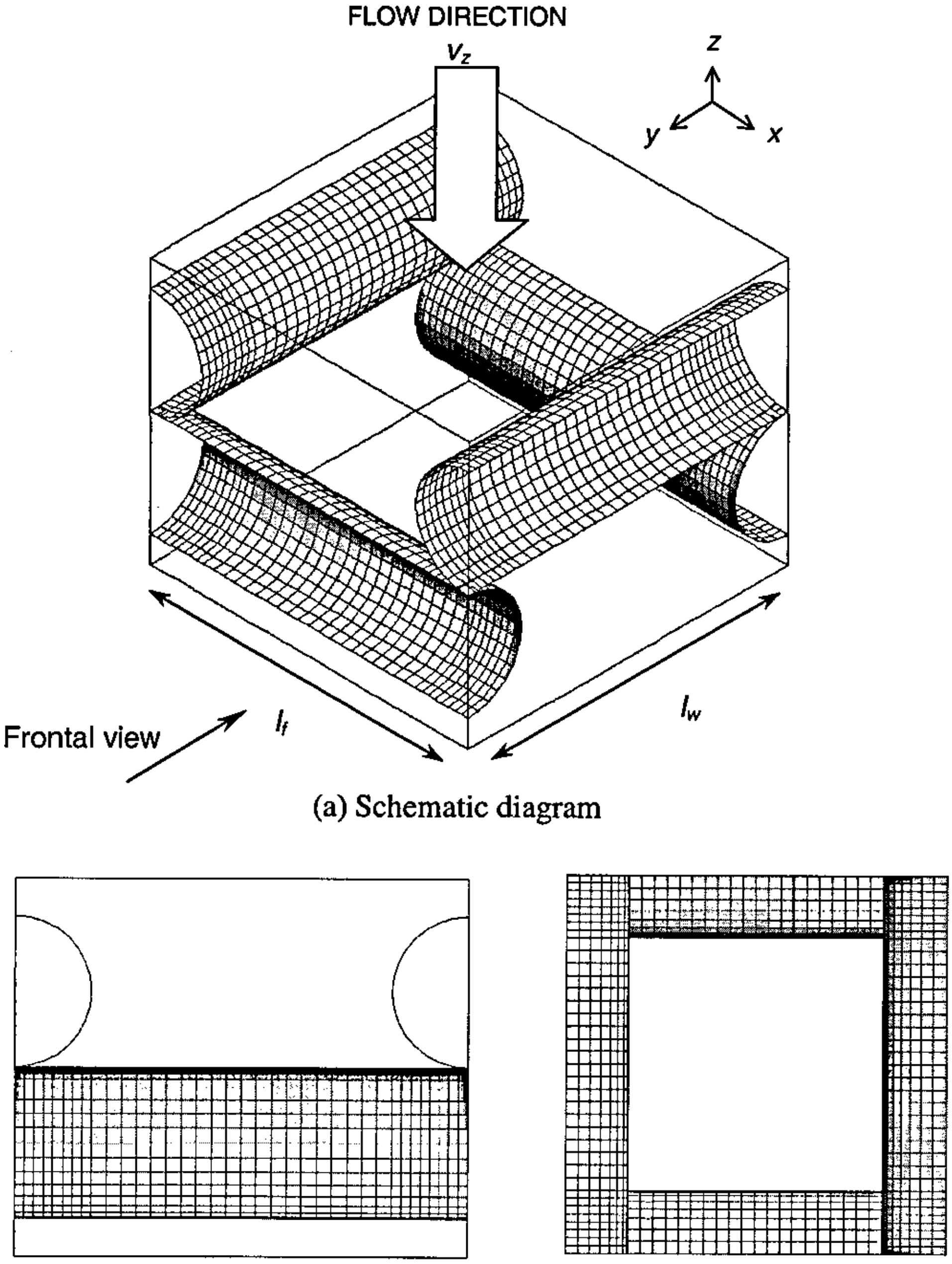
(b) Frontal view

(c) Top view

Figure 5.7 (a) Schematic diagram (b) frontal view and (c) top view of pore type 3 with aperture to diameter ratio 2.0.

(h) **DOMAIN 8**: Pore type 4 with aperture to diameter ratio 2.0.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 6:5:6:2. The mesh contains 5560 elements and 7637 nodes. The crimp diagrams for domain 8 are depicted in Figure 5.8.



(a) Schematic diagram

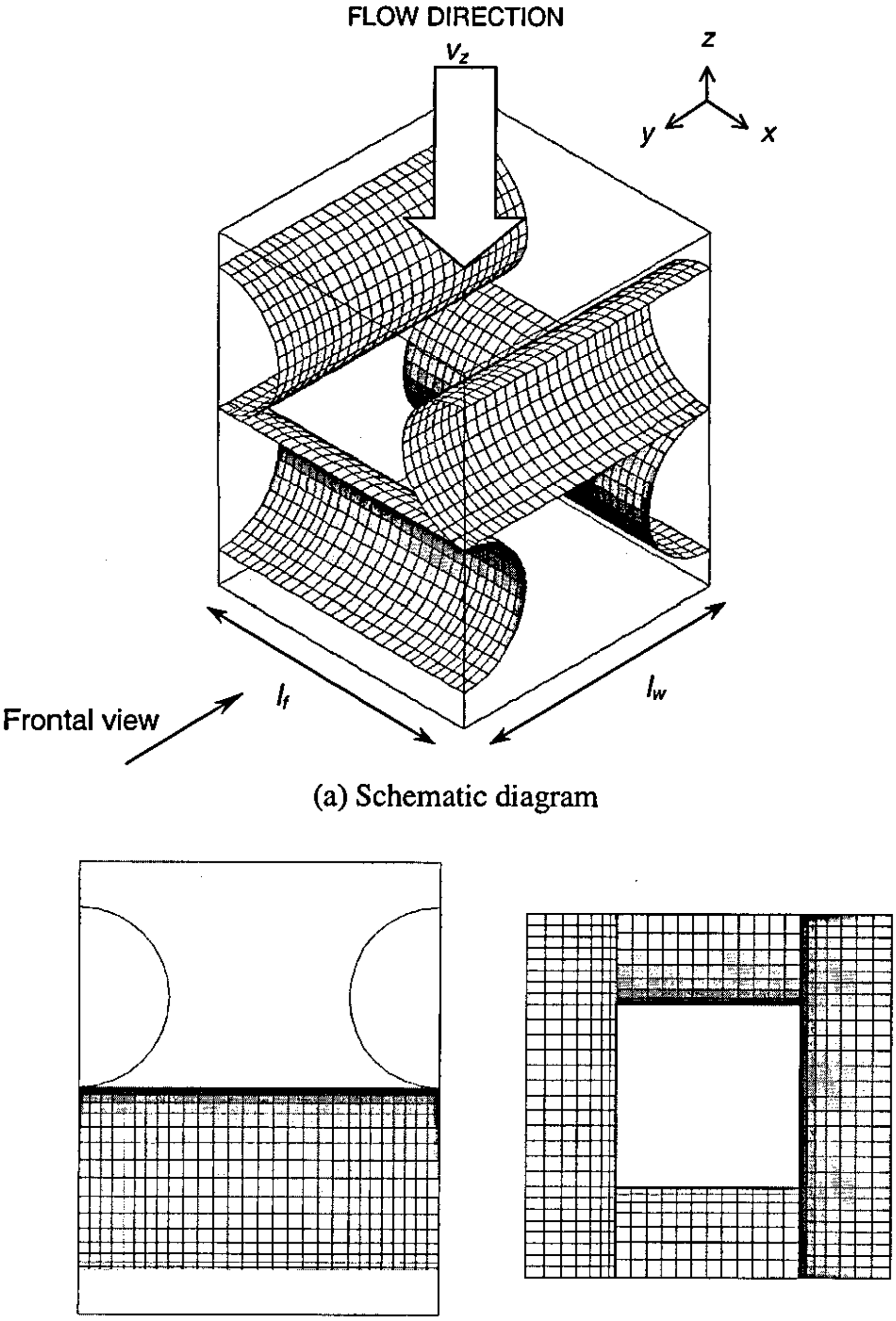
(b) Frontal view

(c) Top view

Figure 5.8 (a) Schematic diagram (b) frontal view and (c) top view of pore type 4 with aperture to diameter ratio 2.0.

(i) **DOMAIN 9**: Pore type 4 with aperture to diameter ratio 2.0.

The ratio of the domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 4:5:4:2. The mesh contains 3768 elements and 5357 nodes. The crimp diagrams for domain 9 are depicted in Figure 5.9.



(a) Schematic diagram

(b) Frontal view

(c) Top view

Figure 5.9 (a) Schematic diagram (b) frontal view and (c) top view of pore type 4 with aperture to diameter ratio 1.0.

Domains 1 to 9 were subsequently discretized into finite element working mesh using COSMOS™ GEOSTAR. An eight-noded linear element of the Lagrange rectangular prisms family was used as the interpolation function for all the developed domains (discussed earlier in section 4.3). An example of the finite element mesh is shown in Figure 5.10 depicting the discretized domain 1.

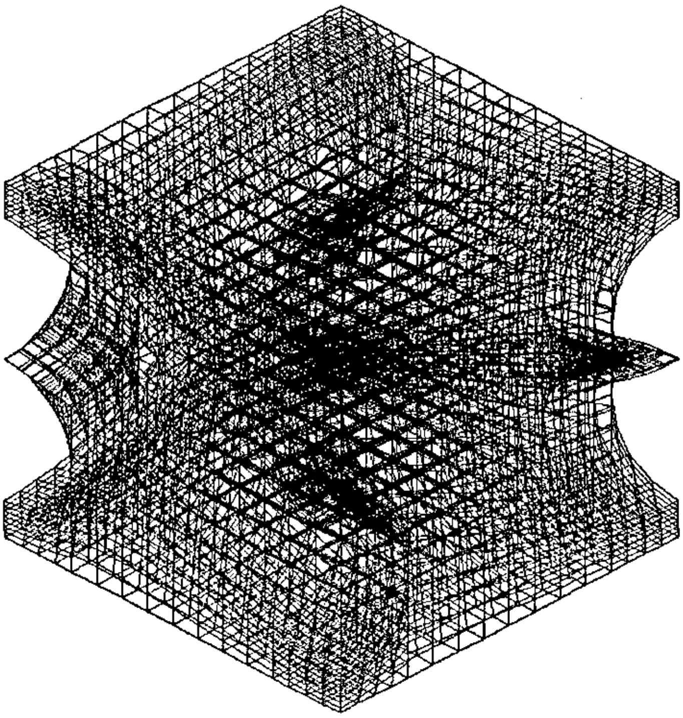


Figure 5.10 Discretized finite element mesh of domain 1.

A summary of the domain setup properties is given in Table 5.1. Based on the selection of the described domains, three parameters are investigated: weave pattern, aperture to diameter ratio or wire spacing, and Reynolds number.

Table 5.1 Summary of the domain setup properties.

Domain	**Pore Type**	***d***	**aperture**	**no. of**	**no. of**	**DOF**	***x*:*y*:*z*:*d***
		(mm)	**(mm)**	**element**	**nodes**		**ratio**
1	Pore Type 1	0.2	0.3	4608	6560	3363	5:5:5:2
2	Pore Type 2	0.2	0.3	4608	6514	3288	5:5:5:2
3	Pore Type 3	0.2	0.3	4608	6527	3282	5:5:5:2
4	Pore Type 4	0.2	0.3	4608	6437	3210	5:5:5:2
5	Pore Type 1	0.2	0.4	5560	7778	3841	6:5:6:2
6	Pore Type 2	0.2	0.4	5560	7722	3760	6:5:6:2
7	Pore Type 3	0.2	0.4	5560	7735	3754	6:5:6:2
8	Pore Type 4	0.2	0.4	5560	7637	3682	6:5:6:2
9	Pore Type 4	0.2	0.2	3768	5357	2770	4:5:4:2

For all the developed domains, the boundary conditions are given as follows. The flow inlet is at the top horizontal surface while the flow exit is at the lower horizontal surface. A range of inlet velocity is used for each case corresponding to a wide range of Reynolds number. An open boundary is taken for the four sidewalls where no boundary condition was specified. Only the pressure boundary conditions are specified at the exit flow for each case. The monofilament cut-sized wires were assumed to be non-permeable and non-slip where velocity was taken to be $v_x = v_y = v_z = 0$.

5.1.2 Physical Properties

A non-Newtonian fluid with the property given in Table 5.2 was used in the simulations. Similar properties were used for the simulations of the Newtonian fluid with n taken to be 1.0.

Table 5.2 Physical properties of the shear thickening fluid used.

Physical Property	**value**
Density, ρ	970 kg m^{-3}
Consistency index, η_0	80.0 kg m^{-1}s^{-1}
Power Law index, n	0.7 - 1.3

A time step of Δt = 0.01s was used throughout the simulations and the θ used for the flow modelling is 0.95.

5.2 Presentation of Results

Simulations were successfully conducted for domains described in earlier sections and the results were presented in the following format. For all cases, the results for the cross sections at x-z plane at $y = l_w/2$ and y-z plane at $x = l_l/2$ are represented graphically using SURFER® 8 commercial software package. The velocity vector $\vec{v} = v_a\hat{i} + v_b\hat{j}$ is used to visualize the fluid flow in the cross-sections. The arrow indicates the combined velocity components in the a- and b-axis of a Cartesian coordinate system where a, b = x, y, z and $a \neq b$. The colour of the arrow indicates the magnitude which is given by $|r| = \sqrt{a^2 + b^2}$, as shown in Figure 5.11. The colour gradient used in each figure may be different as it is automatically determined by the graphics package. Figure 5.12 showed an example of colour gradient used in SURFER® 8.

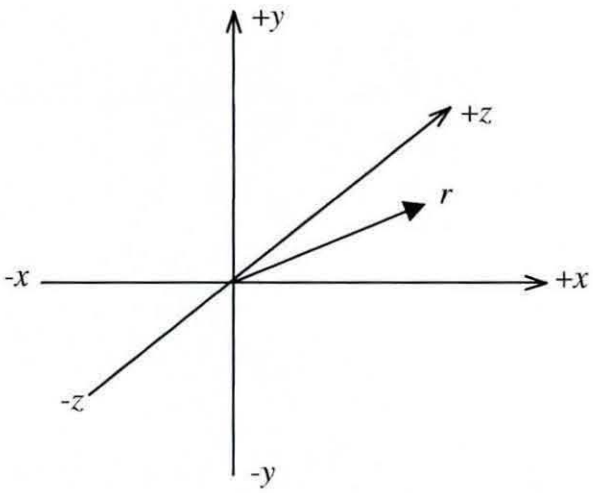


Figure 5.11 Cartesian vector representing x-, y- and z-components of velocity field.

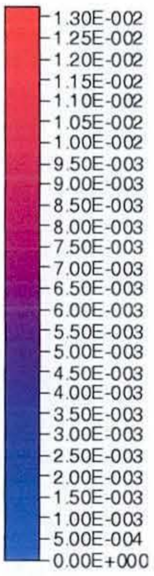


Figure 5.12 The colour legend bar used to represent magnitude of velocity vectors in Surfer 8 graphics software package.

5.3 Validation of Model

Validation is related to the reliability of the mathematical model, while verification addresses the reliability of the approximate solution in comparison with the exact solution of the mathematical problem. Any validation has to assume that the finite element approximation is sufficiently accurate such that there is no interference of the errors in the FE solution, and that the agreement or disagreement with the observed data is due only to the mathematical model. The goal of the computation has to be well specified so that the high quality of the finite element solution can be achieved with respect to this goal. The accuracy of the finite element solution also has to be sufficient with respect to the goal of the analysis (Babuŝka and Strouboulis, 2001).

In this study, three-dimensional test cases were devised to check the computer codes developed before they were applied to the simulations of fluid flow through wire meshes. Three test cases were designed in increasing mesh complexity to progressively test the capability of the algorithm. The developed mathematical model will be tested both qualitatively and quantitatively.

For all the test cases, the velocity v_z corresponds to the approach velocity of the fluid towards the aperture and was set as 0.1 m s^{-1} at the inlet. The boundaries (the cloth filaments) were considered to be impermeable, non-slip surfaces where $v_x = v_y = v_z = 0$. The computational mesh and degrees of freedom (DOF) used for each test case are given in Table 5.3.

Table 5.3 Computational mesh used for each test cases.

Test Case	No. of Elements	No. of Nodes	DOF
1	1920	2511	3544
2	4096	4913	4572
3	4096	4913	4164

A non-Newtonian fluid with the properties given in Table 5.2 was used in the simulations. The power law index n was taken to be 0.9 for all the three test cases. Similar properties were used for the simulations of the Newtonian fluid, but with n taken to be 1.0. Typically three iteration cycles are required to obtain the desired solution that meets the predefined convergence criteria discussed in section 4.6. Depending on the mesh density in the solution domain, the CPU time for each iteration range from 30 min for a coarse mesh to 120 min for a refined mesh. As the U-V-W-P scheme has successfully predicted the fluid flow in all the described domains, the results for continuous Penalty method will not be included as the scheme will only be consulted as secondary comparison for the U-V-W-P results.

5.3.1 Test Case 1: Rectangular Domain

Test Case 1 was used to validate the capability of the program in solving the continuity and motion equations. The main test for the accuracy of the simulation results is ensuring a mass balance over the domain under investigation. A rectangular domain, shown in Figure 5.13, was adapted with four impermeable, no-slip, vertical walls and the inlet was at the upper horizontal surface and the outlet at the bottom horizontal surface. The ratio of the domain length (x), width (y) and height (z) was 3:3:5.

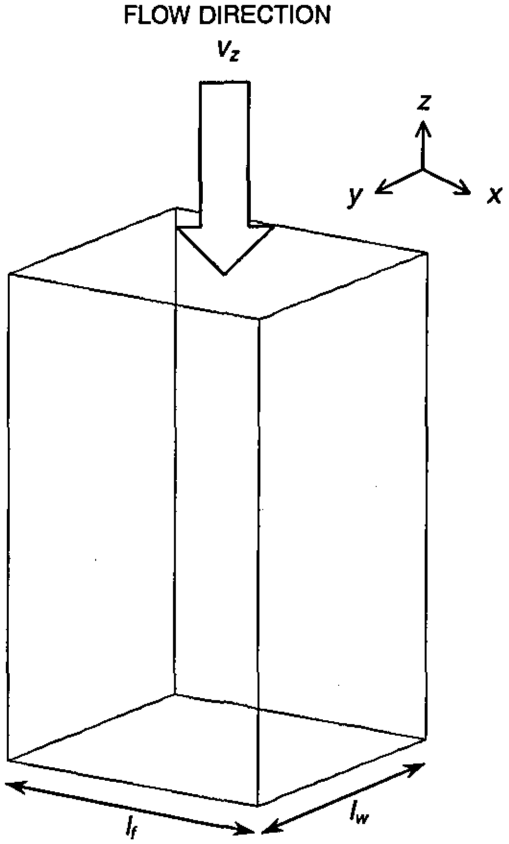


Figure 5.13 A schematic diagram of *Test Case 1.*

Figure 5.14 illustrates a typical velocity vectors of *Test Case 1* in the x-z plane at $y = l_w/2$ for Newtonian and non-Newtonian fluids. The maximum z-direction velocity v_z of non-Newtonian fluid were found to be 4.5% higher that of the Newtonian liquid. This can be explained by the shear thinning effect of the non-Newtonian fluid. The overall mass balance for Newtonian and non-Newtonian simulations is 99.7% and 99.4%, respectively. The excellent accuracy showed that continuity was preserved in the simulations.

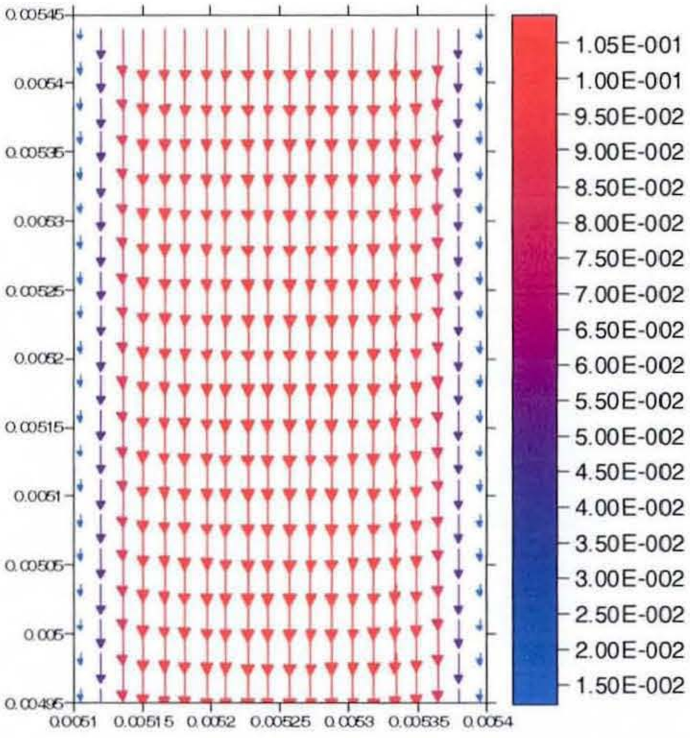


Figure 5.14 The velocity vectors of *Test Case 1* in the *x*-*z* plane at $y = l_w/2$ for non-Newtonian fluid.

5.3.2 Test Case 2: Rectangular Domain with Two Cut-Sized Cylinders

Test case 2 is slightly more complex with two half-cylinders protruding into a rectangular domain, emulating the flow around two parallel fibres that form two sides of an aperture. The ratio of aperture size to wire diameter used in this test case was 1.5 (values from about 1.1 up to about 7 are commonly used ratios in industrial wire meshes). The inlet was again the upper horizontal surface and the outlet was at the bottom horizontal surface. The half cylinder surfaces were considered to be impermeable and non-slip. The ratio of the domain length (*x*), width (*y*), height (*z*) and wire diameter (*d*) was 5:5:2:2 and the schematic computational mesh is shown in Figure 5.15.

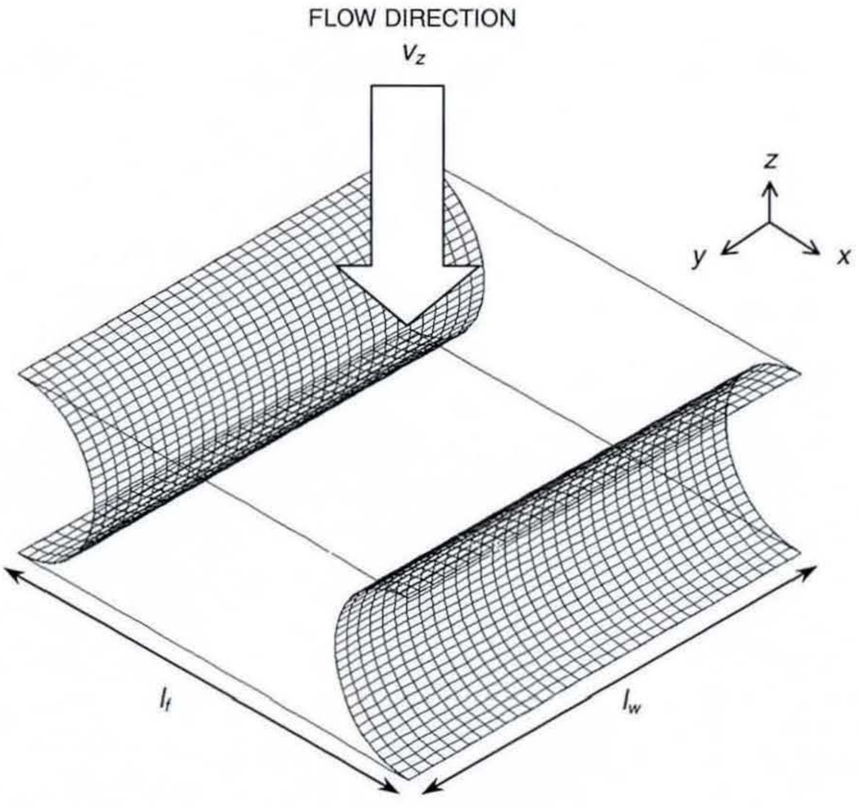
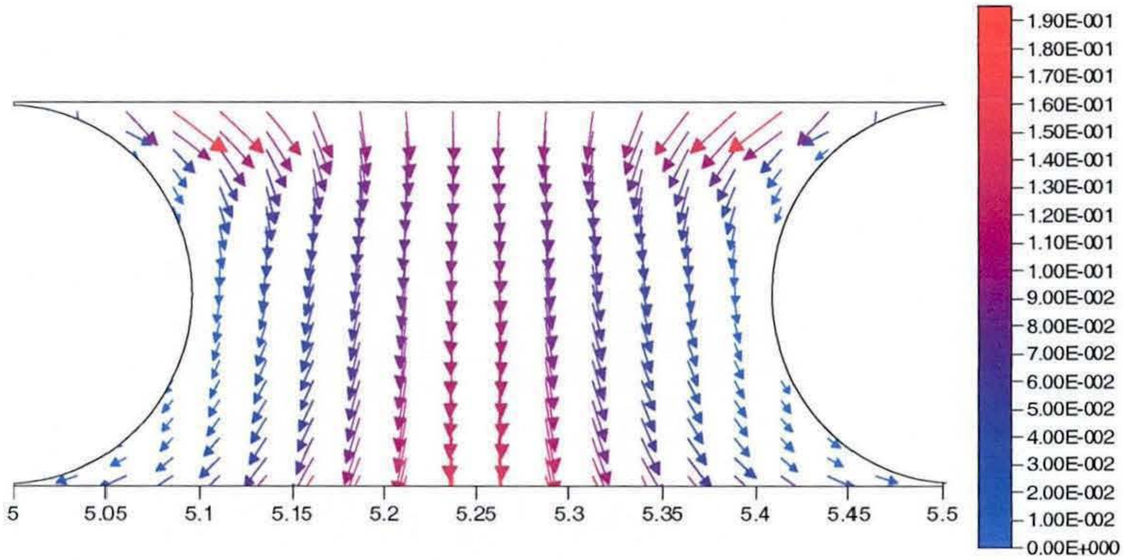
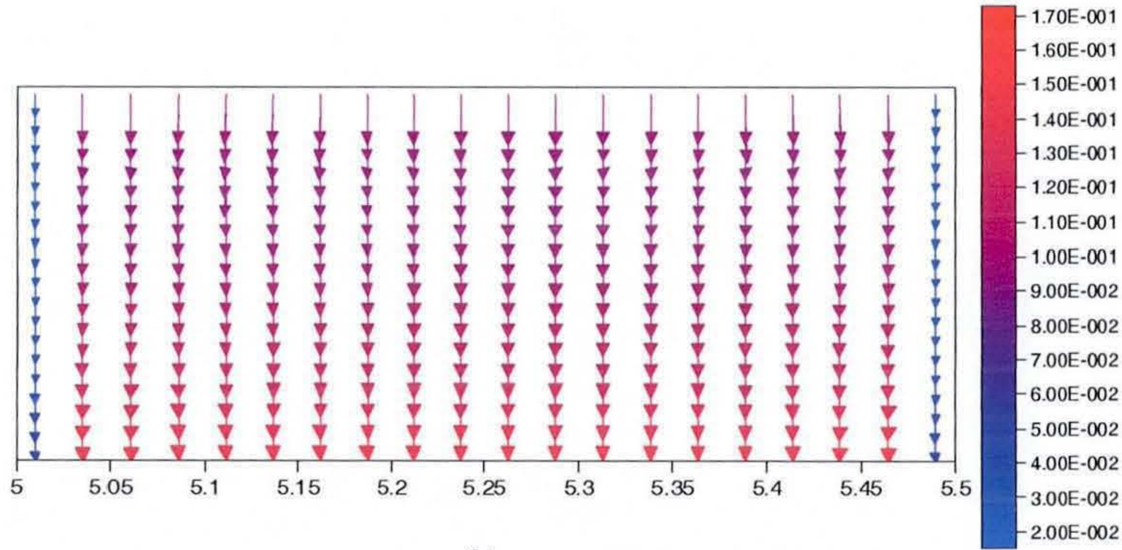


Figure 5.15 A schematic diagram of *Test Case 2*.

Figure 5.16 shows a typical velocity vectors of *Test Case 2* for a non-Newtonian fluid in the *x-z* plane at $y = l_w/2$ and in the *y-z* plane at $x = l_f/2$. The results were obtained after three iteration cycles where negligible changes to the field variables were observed indicating the converged solution has been reached. Higher velocities were observed at the area immediately above the cylinders and at the section between the cylinders compared to the velocity at the inlet and outlet. This is attributed to the compression of the fluid that leads to the acceleration of the fluid velocity.



(a)



(b)

Figure 5.16 The velocity vectors of *Test Case 2* in the (a) x-z plane at $y = l_w/2$ (b) y-z plane at $x = l_f/2$ for non-Newtonian fluid.

The overall mass balance for Newtonian and non-Newtonian simulations in *Test Case 2* showed an outlet flow of 95.0% and 94.8% compared to the inlet flow. The discrepancy is due to the relatively coarse mesh used and can be improved by further mesh refinement. The effect of mesh refinement on the accuracy of the model will be discussed in a later section.

5.3.3 Test Case 3: Rectangular Domain with Four Cut-Sized Cylinders

Test Case 3 has all the basic features of a typical wire mesh domain except that the four supposedly intertwined weft and warp wires were flattened to a horizontal plane to test the response of the model to highly complex geometry and mesh distortion. The four cylinders were jointed together via a 45° slice at each corner. The ratio of the domain length (x), width (y), height (z) and wire diameter (d) was 5:5:2:2. The results from this test case provided powerful insights into understanding the complex flow field in different pore types of a wire mesh. Inlet and outlet settings were similar to *Test Case 1* and *Test Case 2*. The half cylinder surfaces were again considered to be impermeable and non-slip. The schematic domain is shown in Figure 5.17.

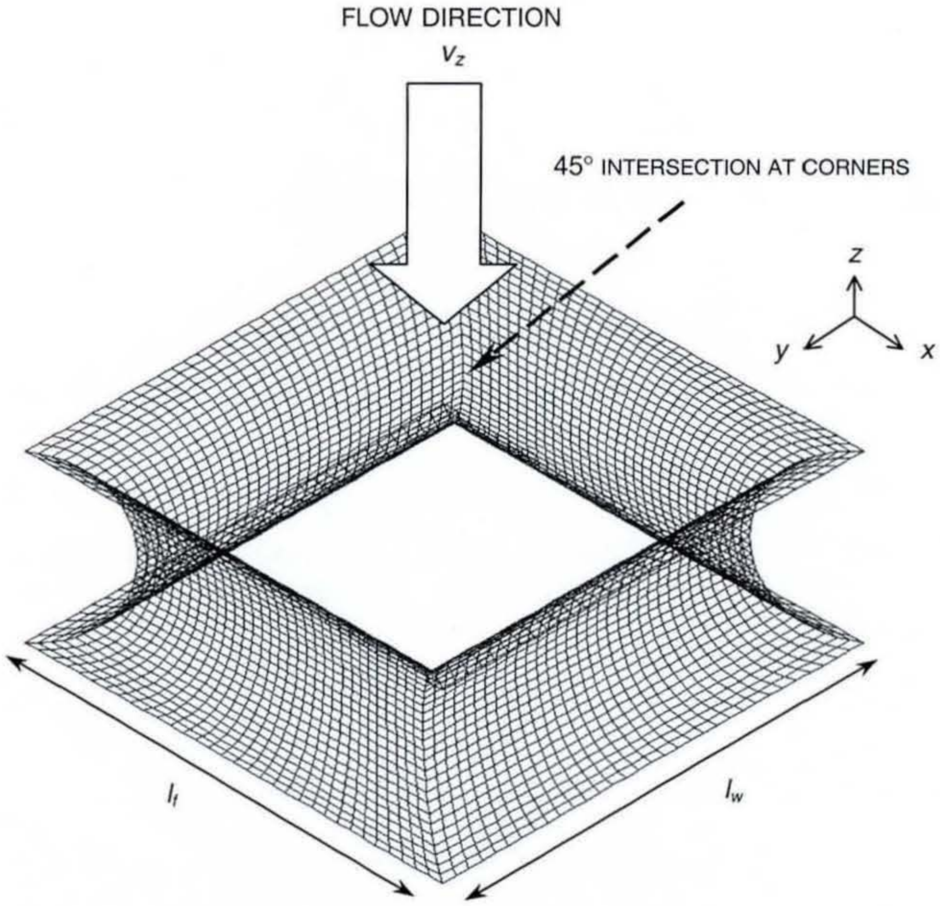
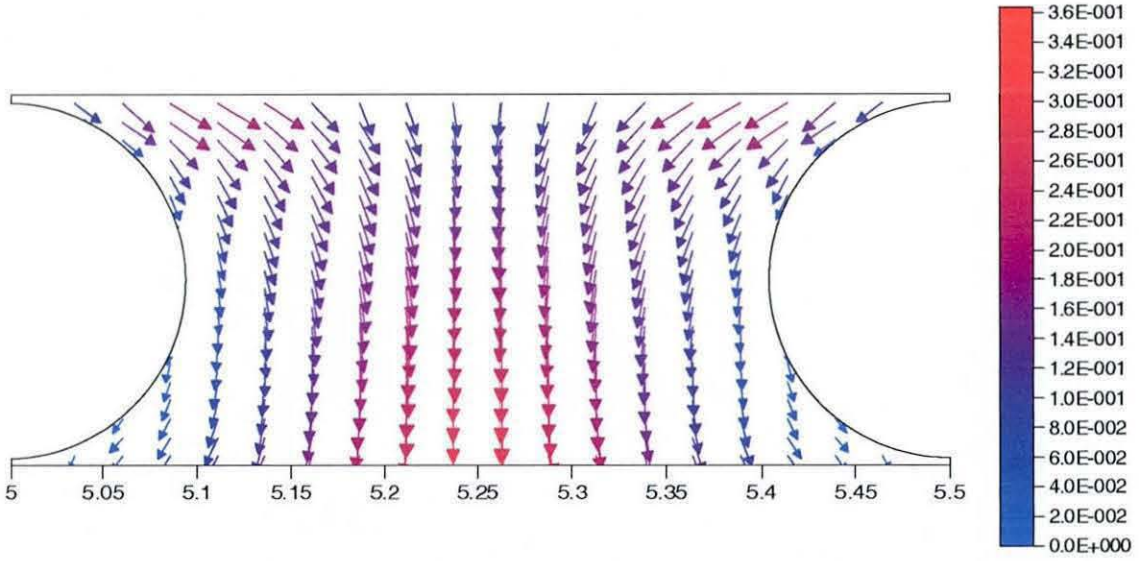
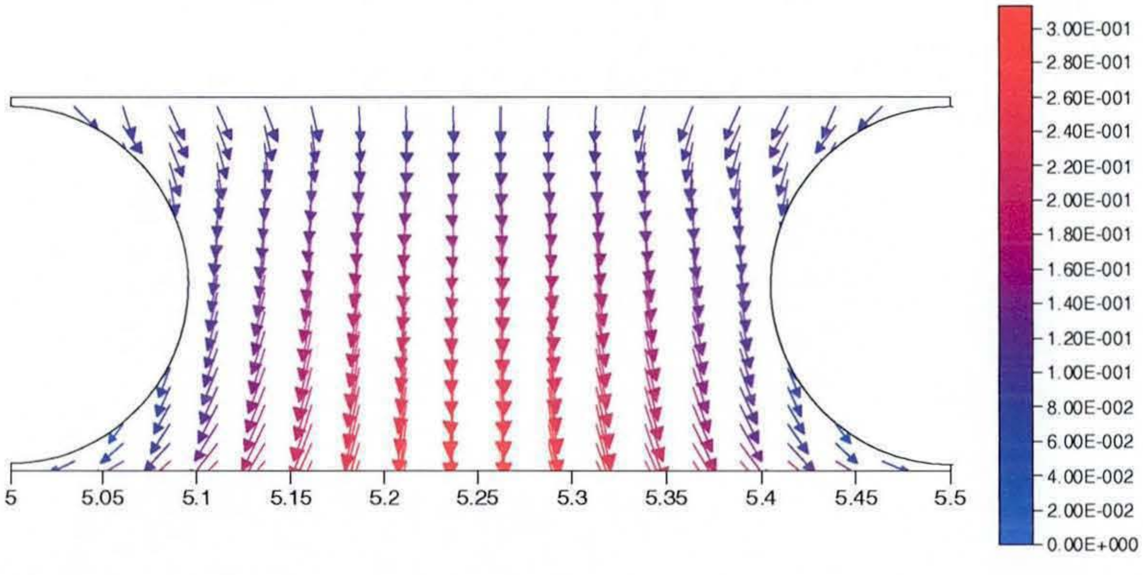


Figure 5.17 A schematic diagram of *Test Case 3*.

Figure 5.18 depicted a typical simulation result of *Test Case 3* for a non-Newtonian fluid in the *x*-*z* plane at $y = l_w/2$ and in the *y*-*z* plane at $x = l_f/2$ The highly complex geometry of the four adjoined cylinders formed a narrow channel for the fluid to flow through. The mass balance for Newtonian and non-Newtonian simulations is 96.1% and 95.8%, respectively. The overall consistency in the results presented for *Test Case 1*, *2* and *3* showed that the model is capable of solving three-dimensional flow problem in a domain with complex geometry.



(a)



(b)

Figure 5.18 The velocity vectors of *Test Case 3* in the (a) *x-z* plane at $y = l_w/2$ (b) *y-z* plane at $x = l_f/2$ for non-Newtonian fluid.

5.3.4 Mesh Refinement

To examine the effect of mesh refinement on the accuracy of the solution, *h*-method refinement has been adopted in this study, as discussed earlier in section 4.7. The elements of the initial mesh were de-refined into larger elements for *Test Case 2* and *Test Case 3*. In *Test Case 2*, a separate simulation run with a very coarse mesh of 512 elements under the same initial and boundary conditions gave a mass balance of 86.2% for Newtonian liquid. In *Test Case 3*, simulation runs using a very coarse mesh of 512 elements gave a mass balance of 86.3% for Newtonian fluid. These results gave clear evidence that the model accuracy can be improved by mesh refinement. The quantitative comparison on the simulation results due to mesh refinement was summarized in Table 5.4.

Table 5.4 Comparison of mass balance at different level of mesh refinement.

	no. of elements	no. of nodes	DOF	% Mass balance
Test Case 2	512	729	1227	86.2
	4096	4913	4572	95.0
Test Case 3	512	729	1011	86.3
	4096	4913	4164	96.1

It should be reiterated that from practical experience gained in this study, the effort of increasing mesh density is not a trivial matter for a three-dimensional domain. It is not unusual for the number of elements to escalate in an order of magnitude at each level of mesh refinement. Hence, a strategy of de-refinement has been used instead of refinement to achieve the purpose of examining the mesh refinement effect on the accuracy of the model.

5.4 Results from Domain 1: Pore Type 1 with Aperture/Diameter Ratio of 1.5

Typical velocity vectors for the flow of Newtonian fluid, shear thickening fluid and shear thinning fluid through domain 1 in the *x-z* plane at $y = l_w/2$ and *y-z* plane at $x = l_f/2$ are illustrated in Figures 5.19 - 5.24. For each case, the flow is observed to swirl inwards towards the aperture on both *x-z* and *y-z* planes corresponding to the location of the wire mesh on both planes. Higher velocities are also observed at the area immediately above the wire mesh and in the section between the wires, compared to the velocity at the inlet and outlet. The cross sections of domain 1 is symmetrical at $y = l_w/2$ and $x = l_f/2$. This is reflected in the velocity distributions and the pressure contour shown in Figures 5.19 - 5.24.

It is observed that as the power law index increases from 1.0 to 1.3, there is a corresponding rise in the simulated pressure drop value. This can be attributed to the shear thickening effect. The opposite is true when the power law index decreases from 1.0 to 0.7 where the pressure drop decreases due to the shear thinning effect.

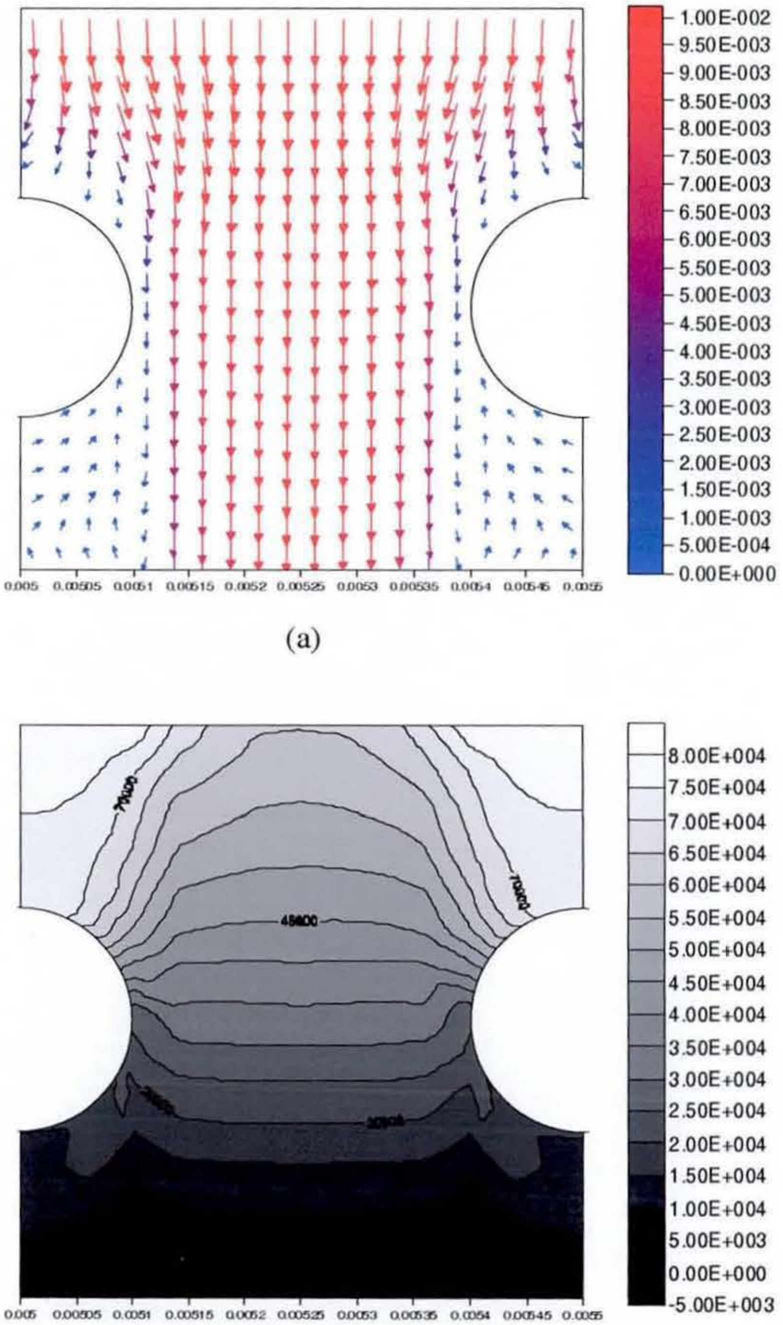
To describe the flow around the wire mesh, the approach used by Wieghardt (1953) was adopted. The drag coefficient defined by equation (2.6) was used. The Reynolds number defined by equation (2.7) however has to be modified in consideration of the non-Newtonian fluid used in the current study. Chhabra and Richardson (1985) used a modified Reynolds number for a non-Newtonian fluid as defined in equation (2.12), a form similar to that used by Metzner (1956) and Skelland (1967) with porosity taken into consideration. In their paper, however, they erroneously took *d* to be the wire diameter rather than the hydraulic diameter,

d_h, which can be taken as the aperture side dimension in the case of a square mesh. A more appropriate correlation for Re_{NN} consistent to that used by earlier researchers is therefore given as:

$$\mathrm{Re}_{NN} = \frac{\rho(u/\varepsilon)^{2-n} d_h^{\,n}}{\eta_0} \tag{5.1}$$

where Re_{NN} reduces to the Reynolds number for a Newtonian liquid when $n = 1$.

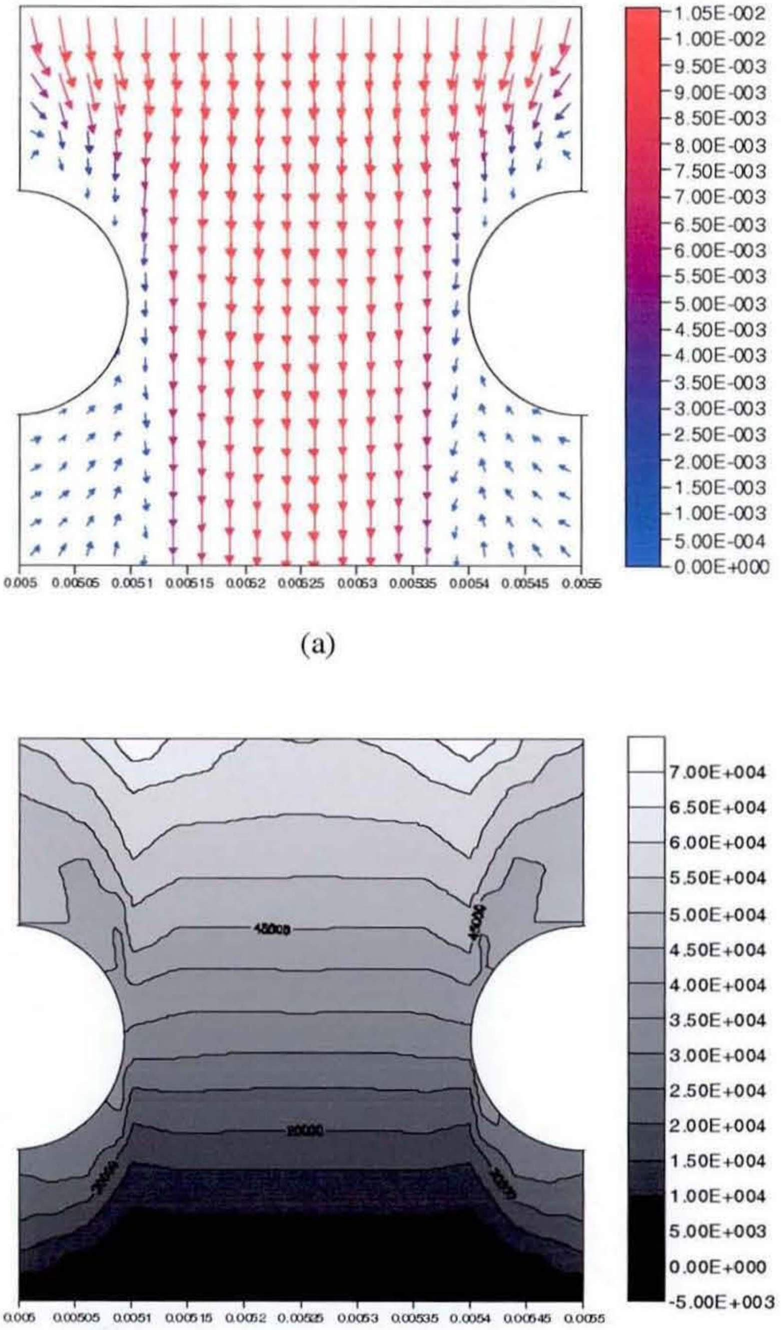
For the simulations in this work, the relationship between the flow rate and the pressure drop for each fluid is calculated from the model and subsequently presented using equations (2.6) and (5.1). The drag coefficient versus Reynolds number plot for domain 1 corresponding to various power law indices are presented in Figure 5.25.



(a)

(b)

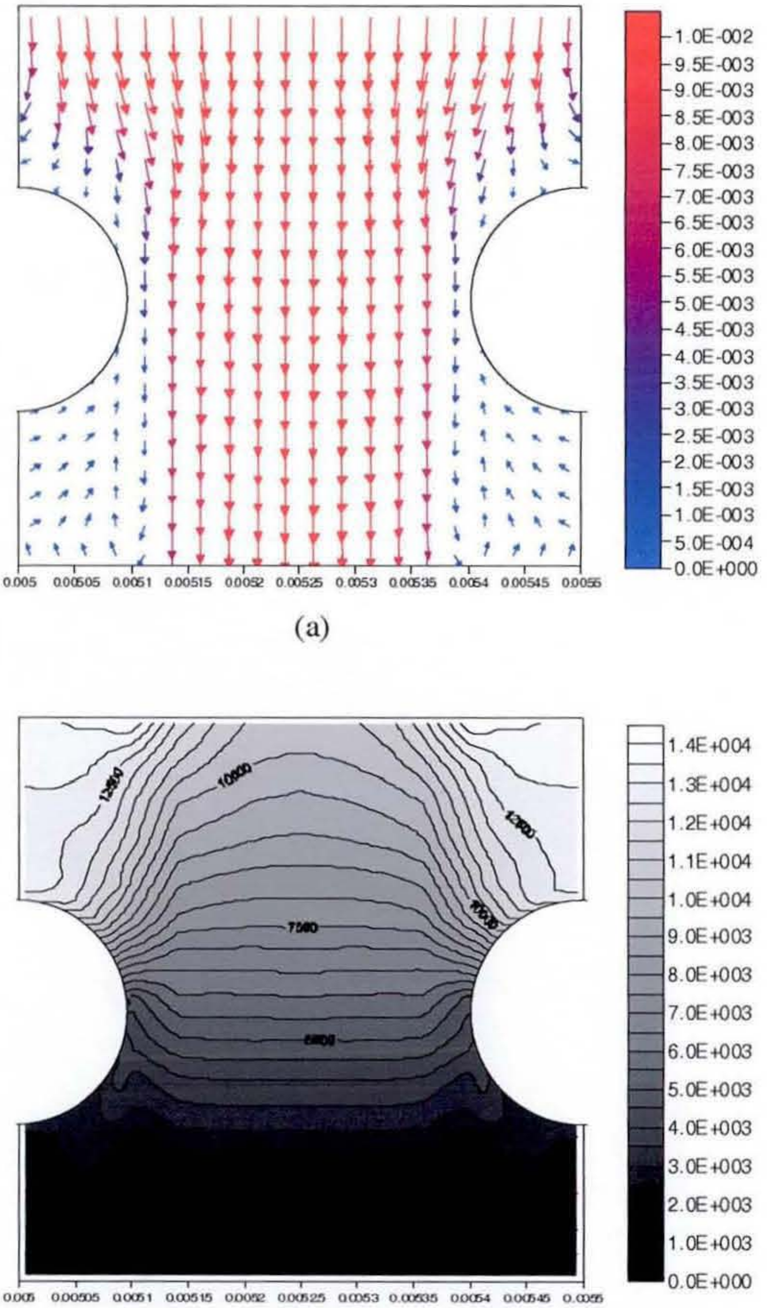
Figure 5.19 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 1 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

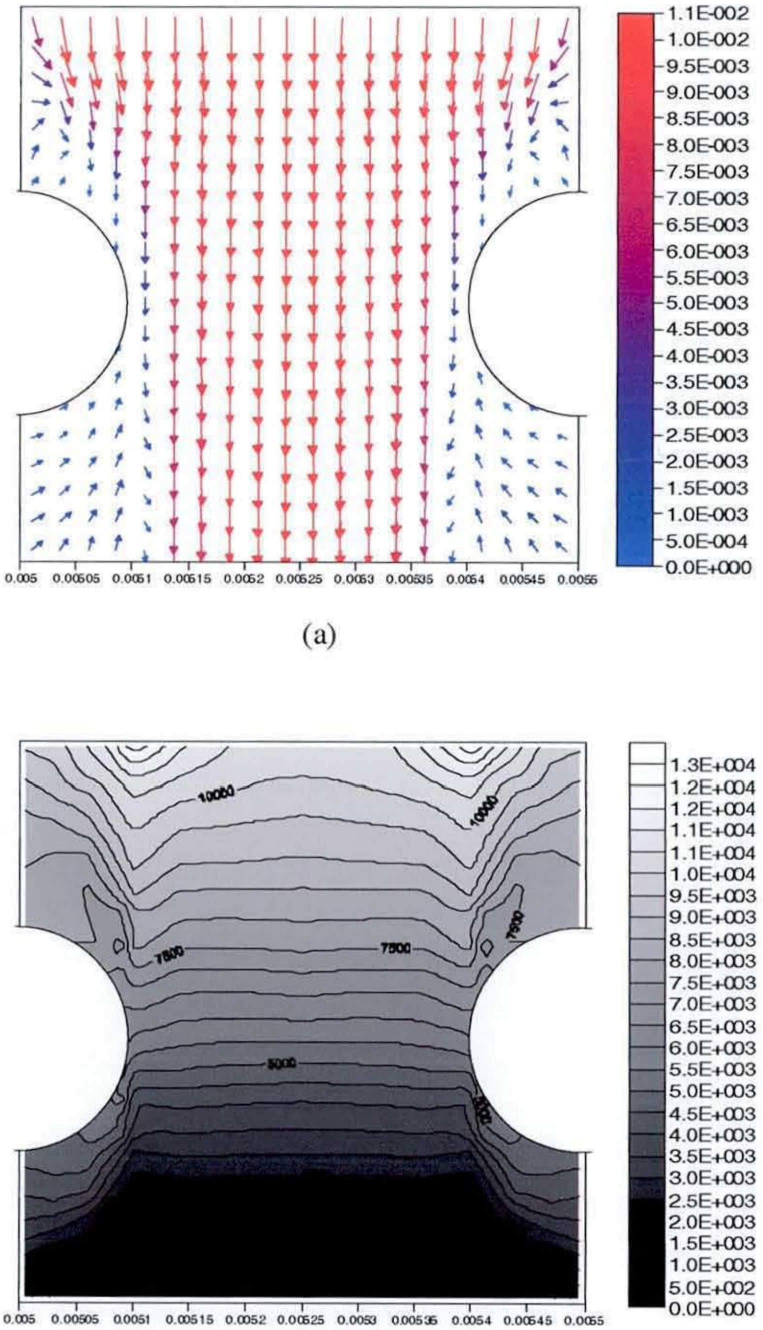
Figure 5.20 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 1 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

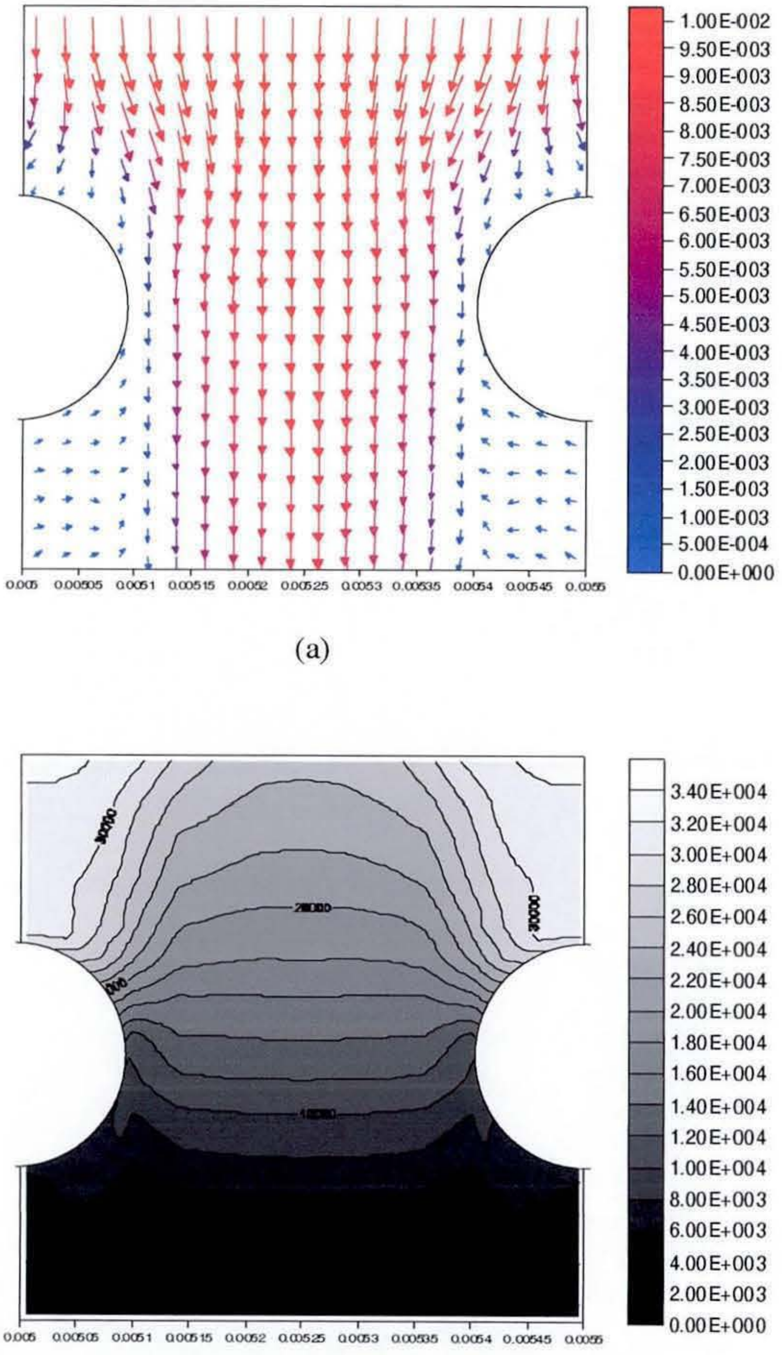
Figure 5.21 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 1.1$) flow through domain 1 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

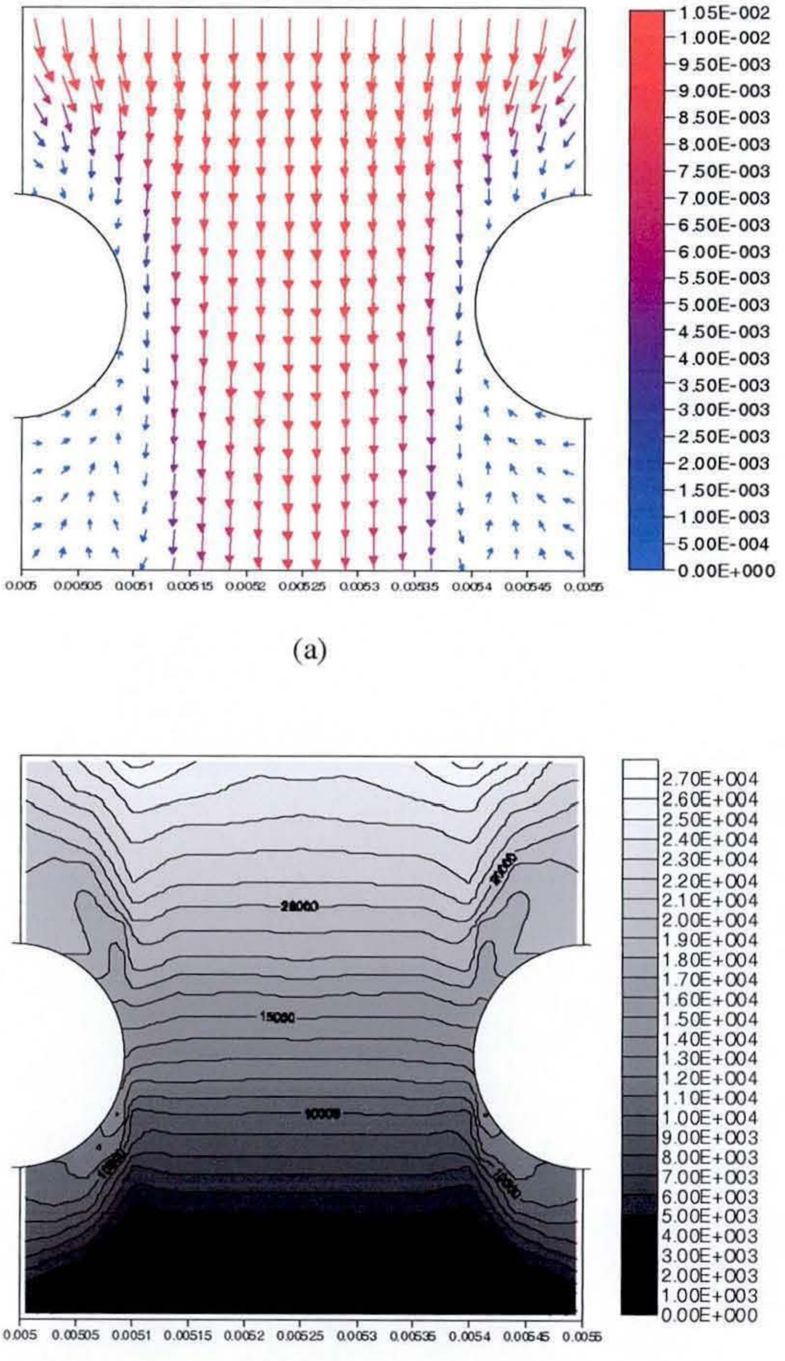
Figure 5.22 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 1.1$) flow through domain 1 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.23 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 1 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.24 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 1 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

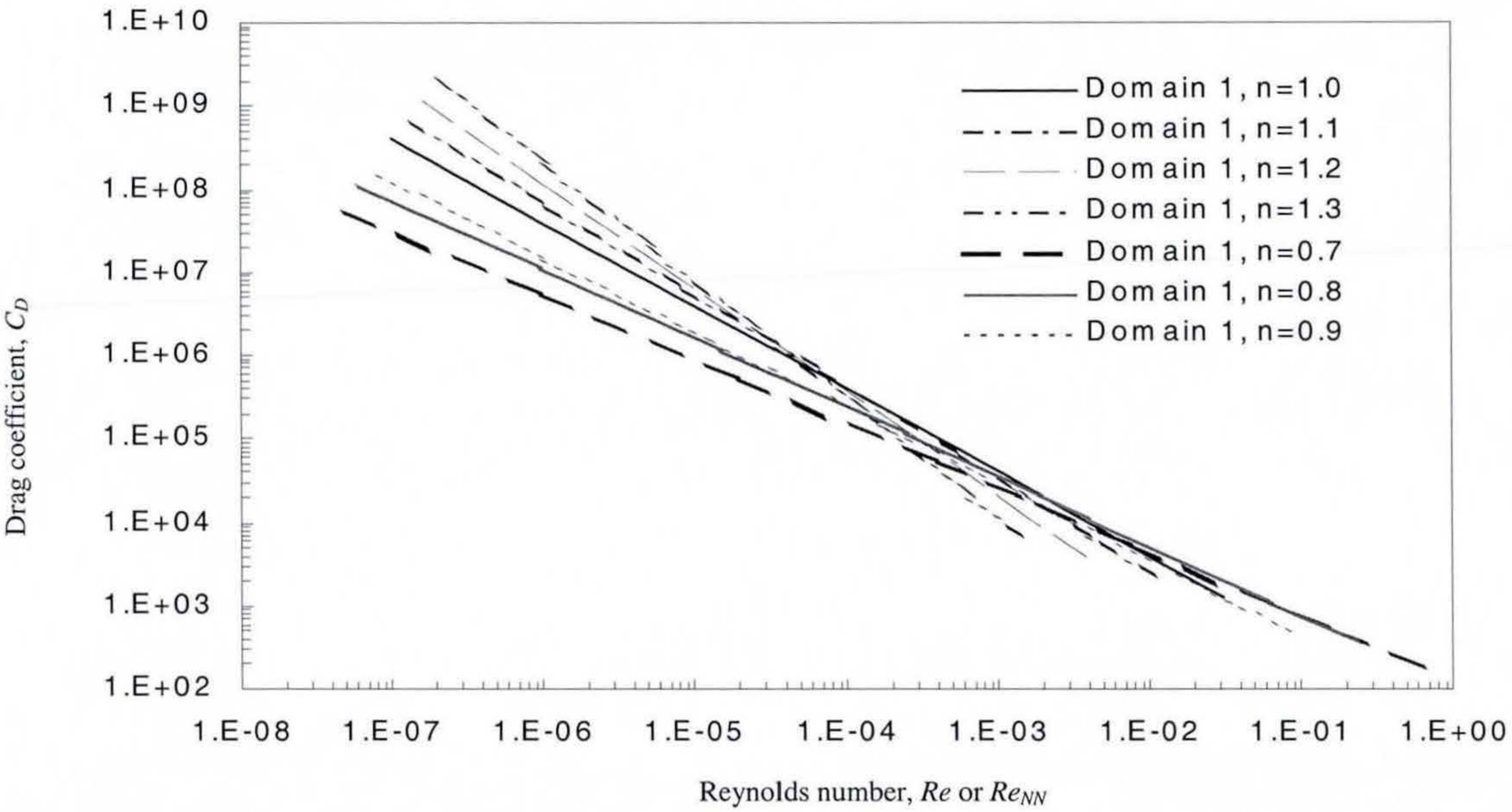
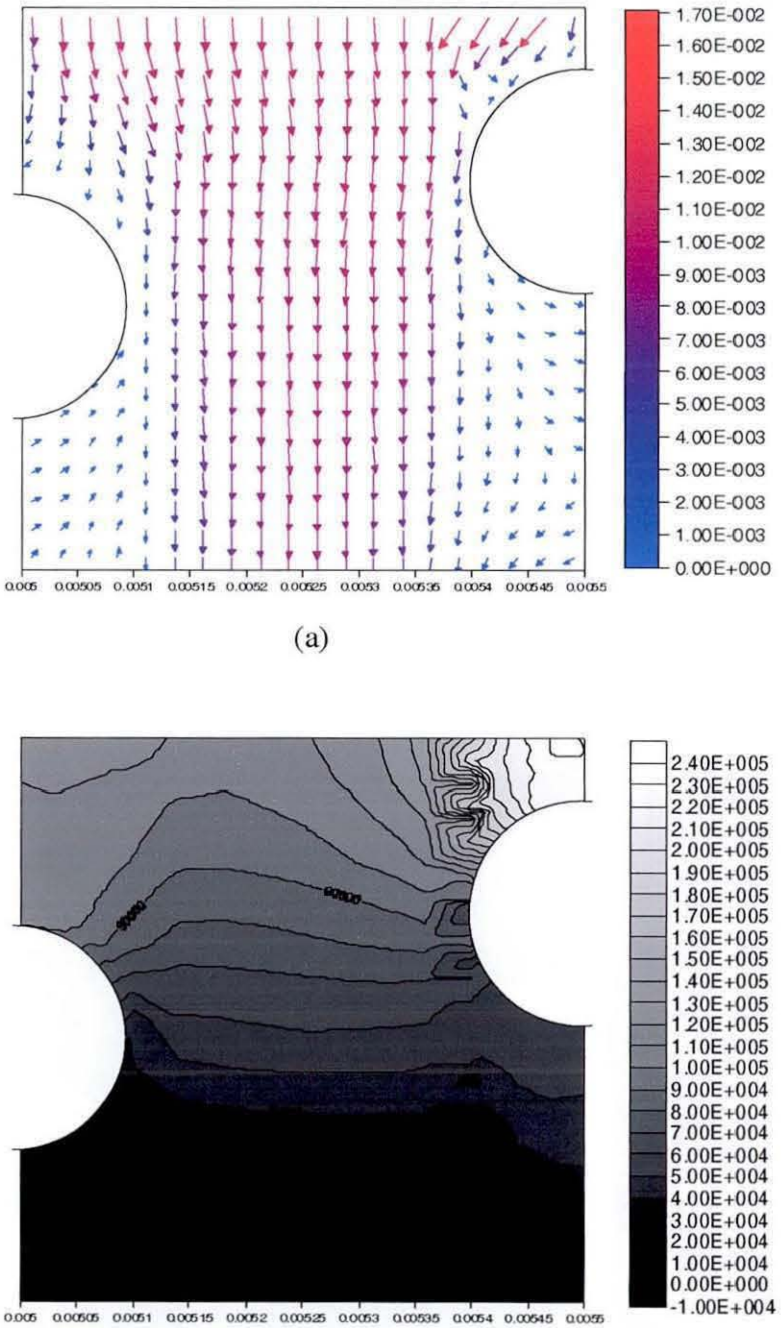


Figure 5.25 Drag coefficient *vs* Reynolds number for Newtonian and non-Newtonian fluid flow through domain 1.

5.5 Results from Domain 2: Pore Type 2 with Aperture/Diameter Ratio of 1.5

Typical velocity vectors for the flow of Newtonian fluid, shear thickening fluid and shear thinning fluid through domain 2 in the *x-z* plane at $y = l_w/2$ and *y-z* plane at $x = l_f/2$ are illustrated in Figures 5.26 - 5.31. Pore type 2 has the most unsymmetrical geometry among the four basic pore types. This is evident in the plots of velocity and pressure illustrated in Figures 5.26 - 5.31 where an absence of symmetrical pattern in both planes is observed. The flow is again observed to swirl inwards towards the aperture on both *x-z* and *y-z* planes corresponding to the location of the wire mesh on both planes. However, the velocity above the right wire mesh in the *x-z* plane plots is higher than that above the left wire mesh. The corresponding cross sections plots in the *y-z* plane at $x = l_f/2$ showed the complex interaction of flow as fluid flow past pore type 2.

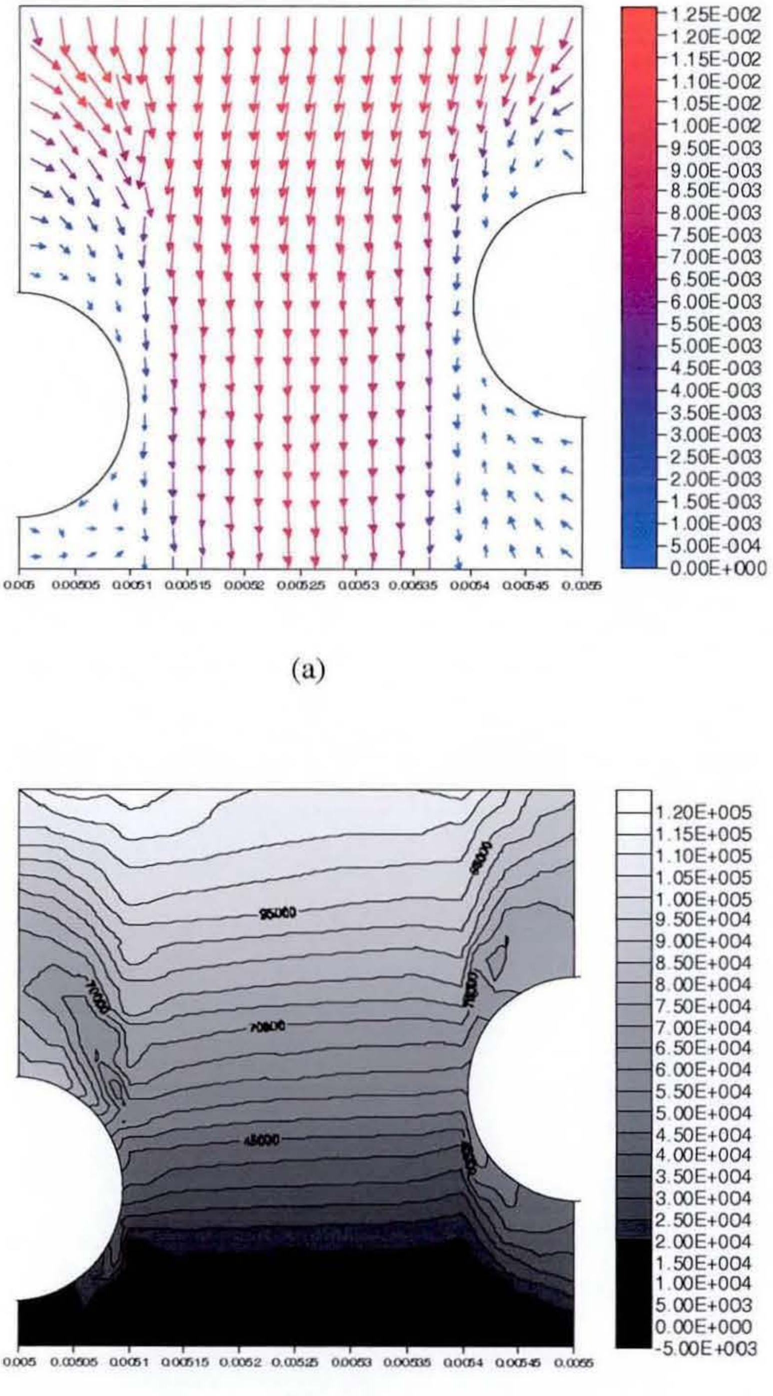
As in the case of domain 1, it is observed that the as the power law index increases from 1.0 to 1.3, there is a corresponding rise in the simulated pressure drop value. This can be attributed to the shear thickening effect. The opposite is also true when the power law index decreases from 1.0 to 0.7 where the pressure drop decreases due to the shear thinning effect. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.32.



(a)

(b)

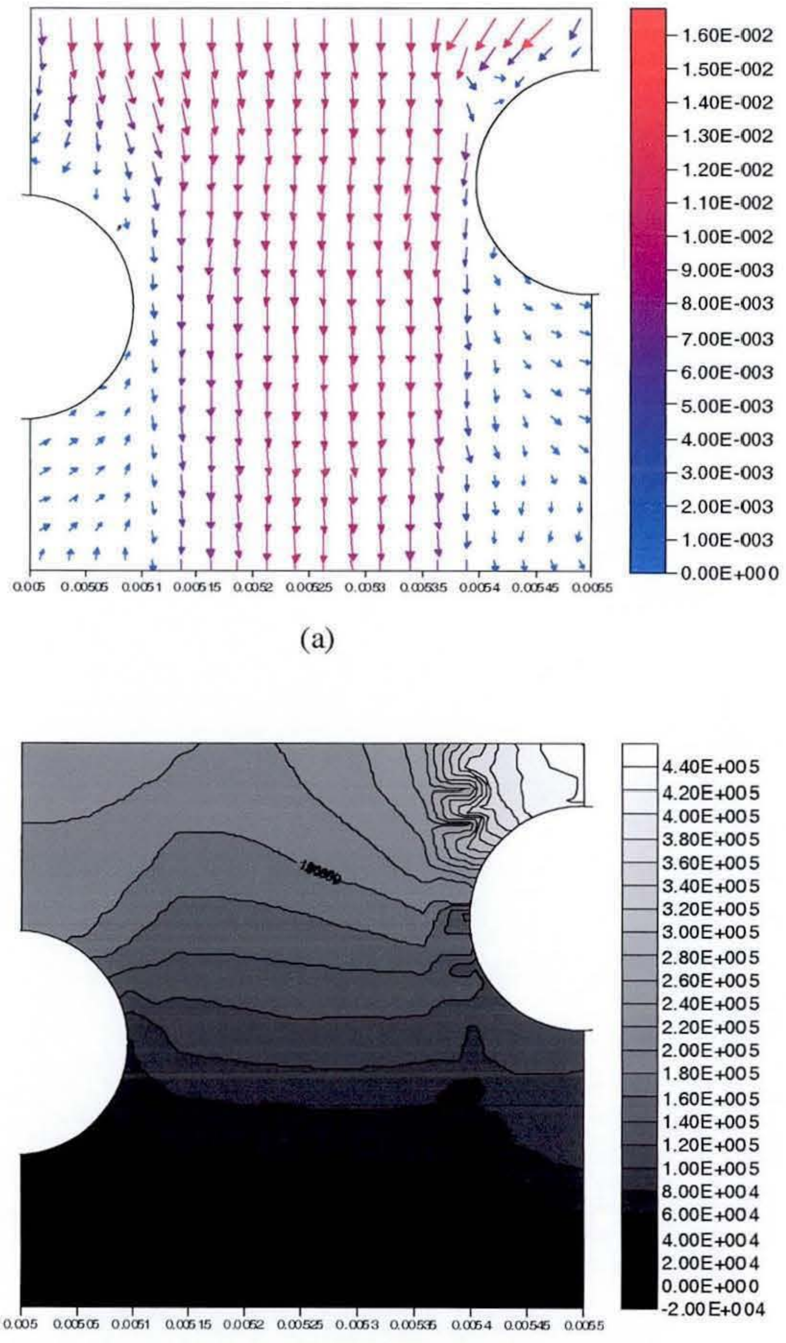
Figure 5.26 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 2 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

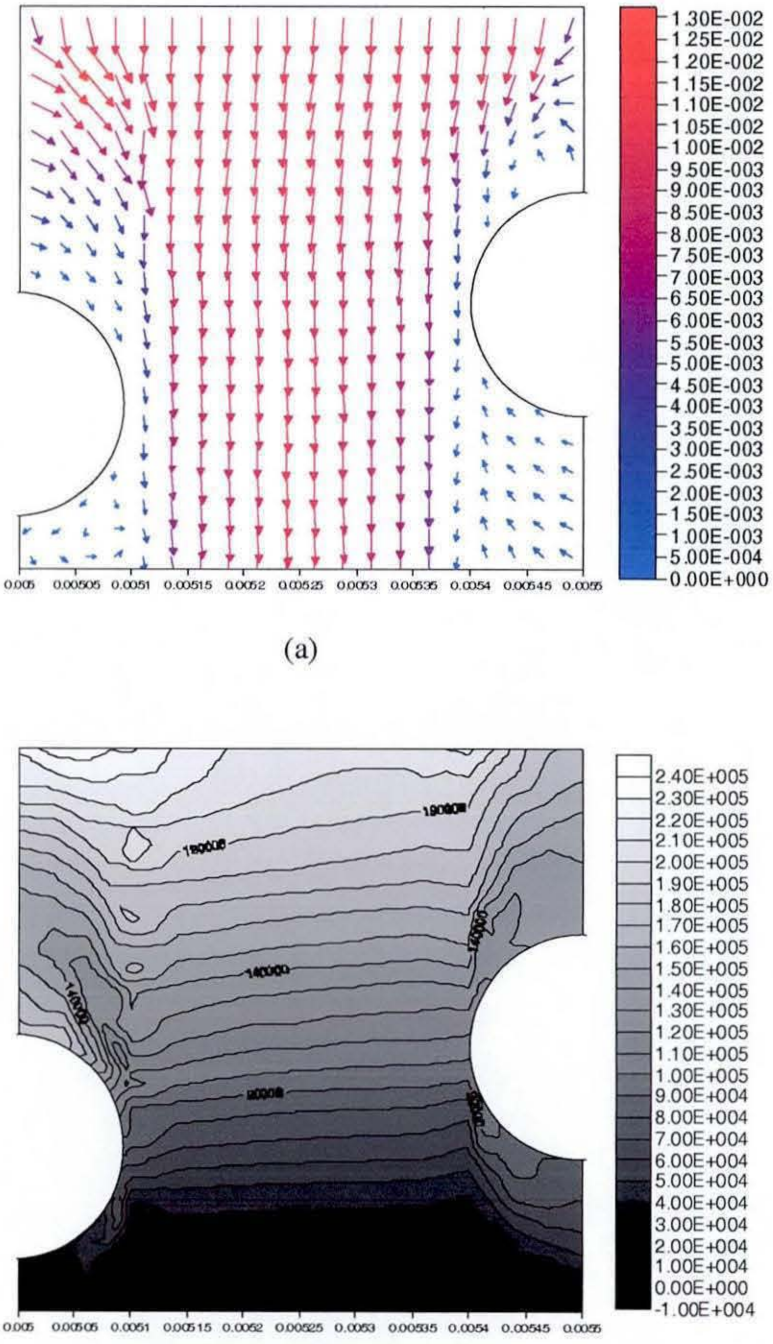
Figure 5.27 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 2 in the *y*-*z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

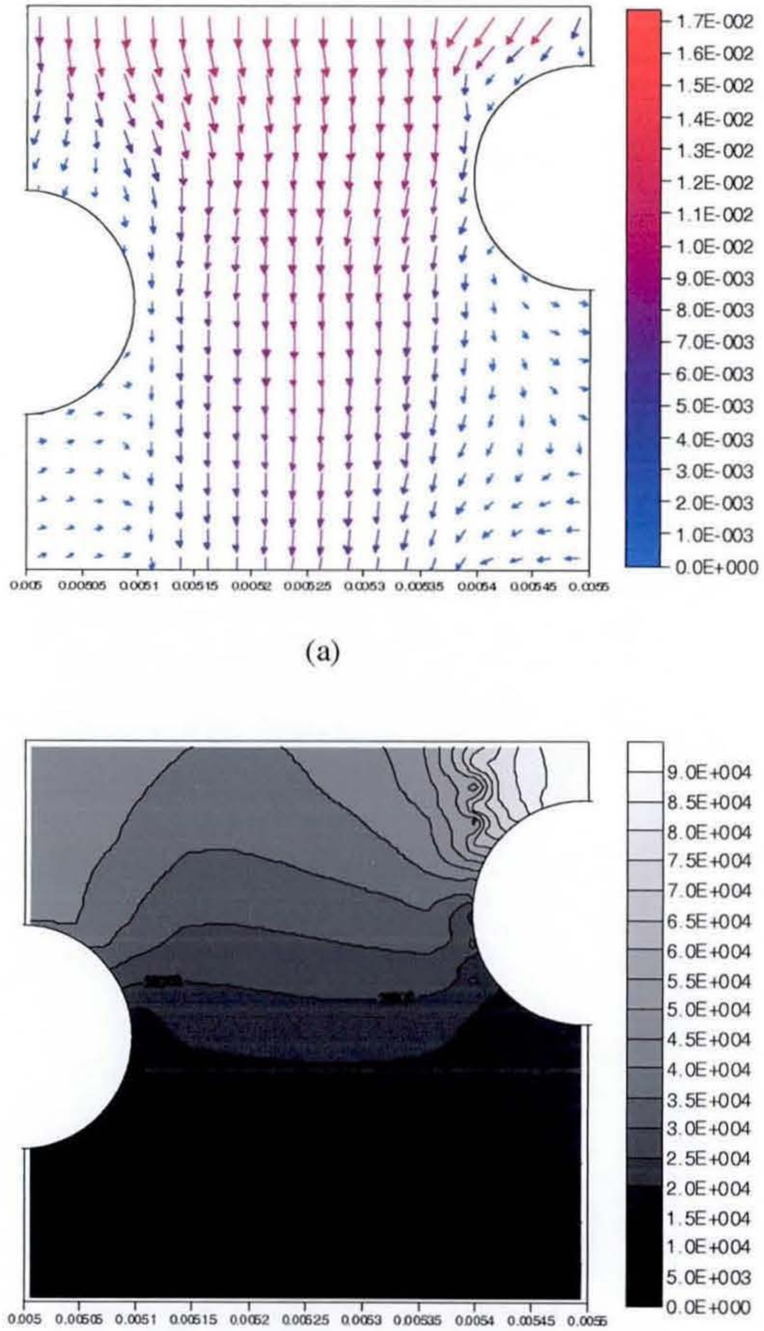
Figure 5.28 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 1.1$) flow through domain 2 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

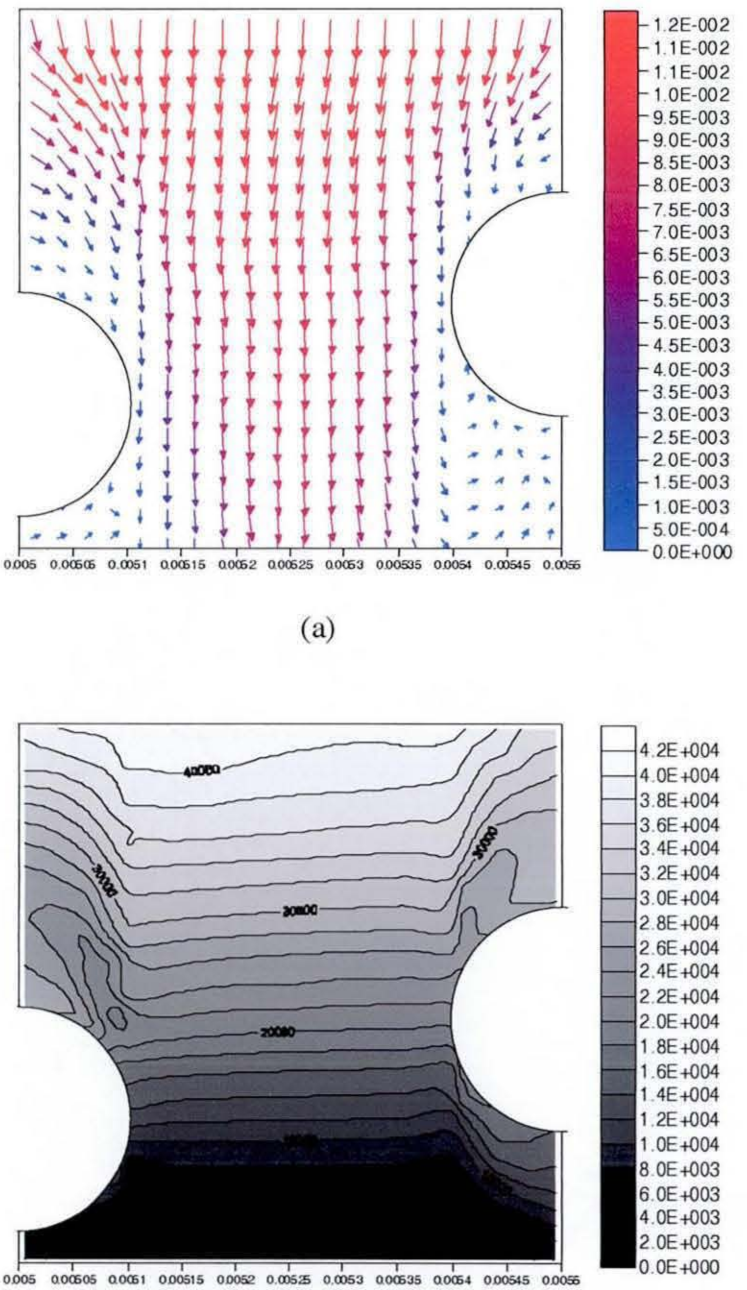
Figure 5.29 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 2 in the y-z plane at $x = l_f/2$ with inlet velocity v_z = -0.01ms^{-1}.



(a)

(b)

Figure 5.30 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 2 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.31 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 0.9) flow through domain 2 in the y-z plane at $x = l_f/2$ with inlet velocity v_z = -0.01ms^{-1}.

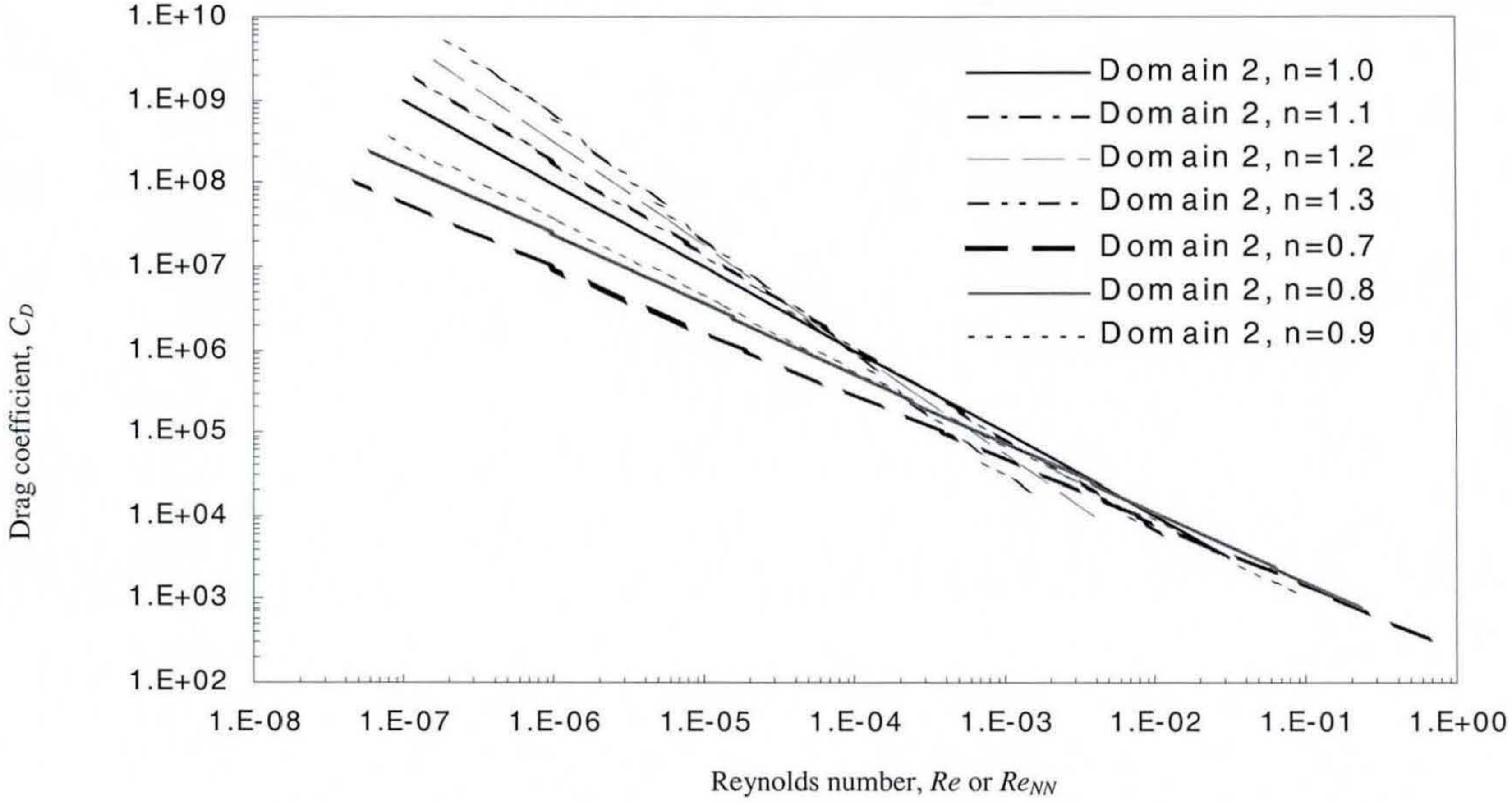
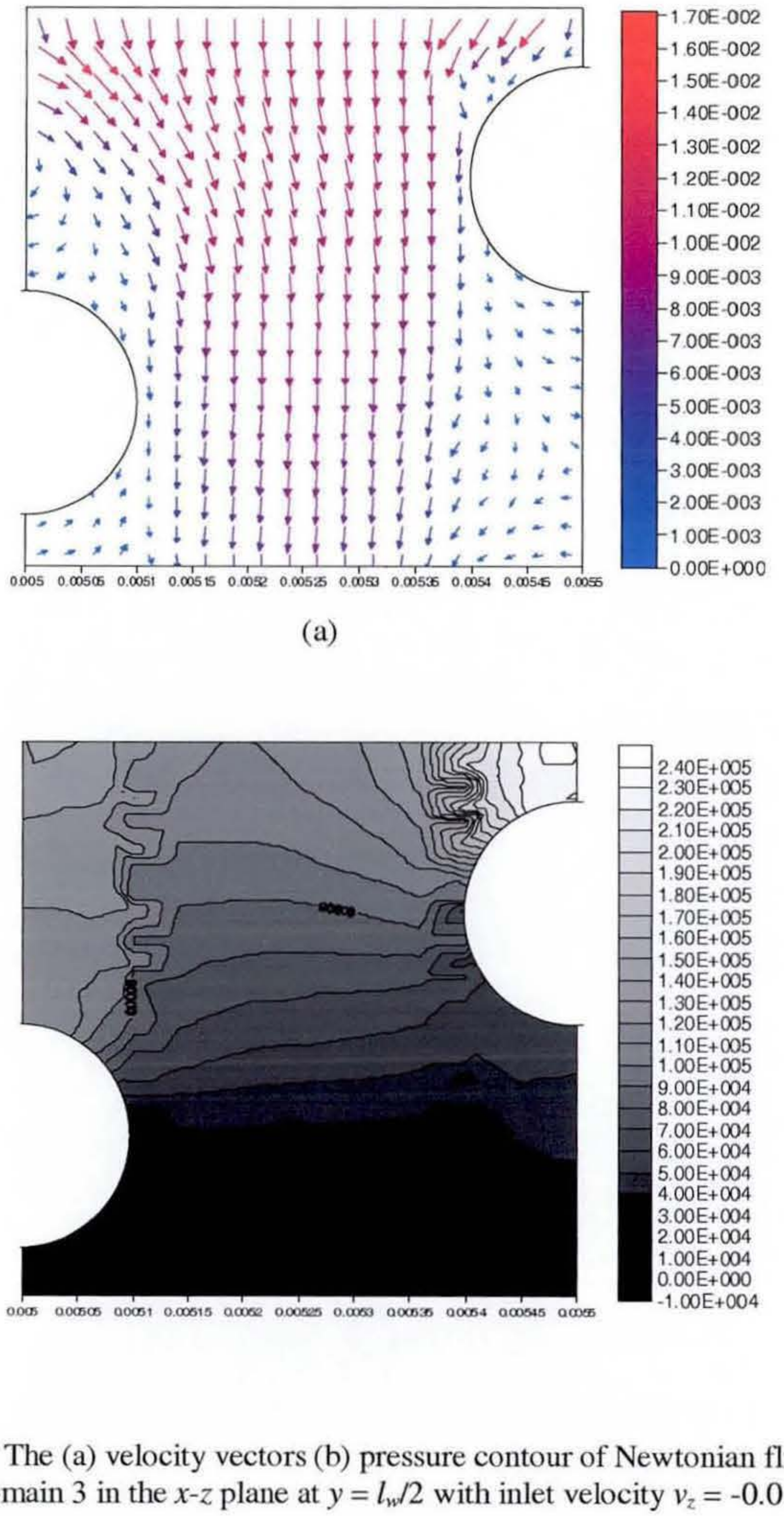


Figure 5.32 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 2.

5.6 Results from Domain 3: Pore Type 3 with Aperture/Diameter Ratio of 1.5

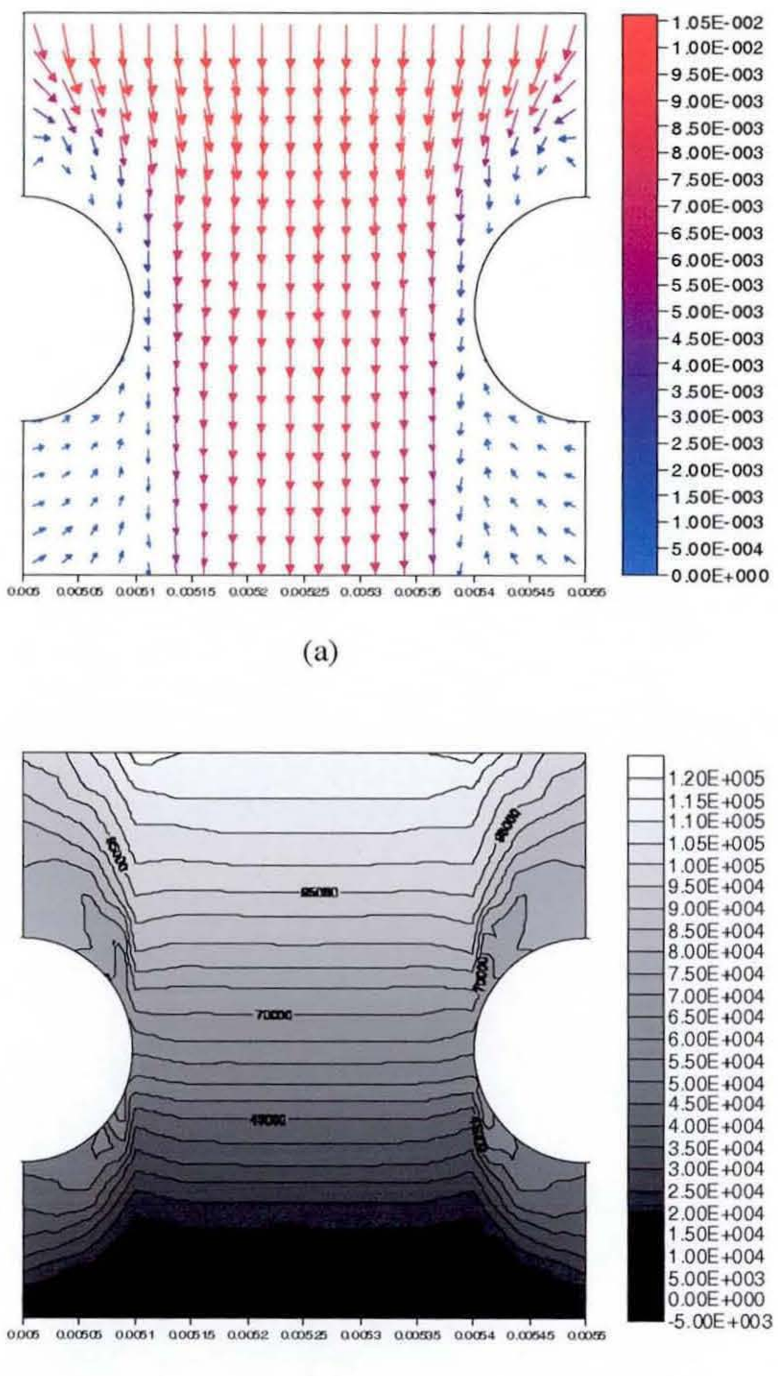
Typical velocity vectors for the flow of Newtonian fluid, shear thickening fluid and shear thinning fluids through domain 3 in the *x-z* plane at $y = l_w/2$ and *y-z* plane at $x = l_f/2$ are illustrated in Figures 5.33 - 5.38. Pore type 3 has a symmetrical geometry in the *y-z* plane but not in the *x-z* plane. This is reflected in the *y-z* plane plot given in Figures 5.34, 5.36, 5.38 and *x-z* plane plots given in Figures 5.33, 5.35, 5.37. The pattern of the fluid flow in the *x-z* plane corresponding to the position of the wire mesh is similar to that observed in domain 2 while flow pattern in the *y-z* plane is similar to that shown for domain 1.

As in the case of domain 1 and domain 2, it is observed that as the power law index increases from 1.0 to 1.3, there is a corresponding rise in the simulated pressure drop value while the pressure drop decreases when the power law index decreases from 1.0 to 0.7. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.39.



(a)

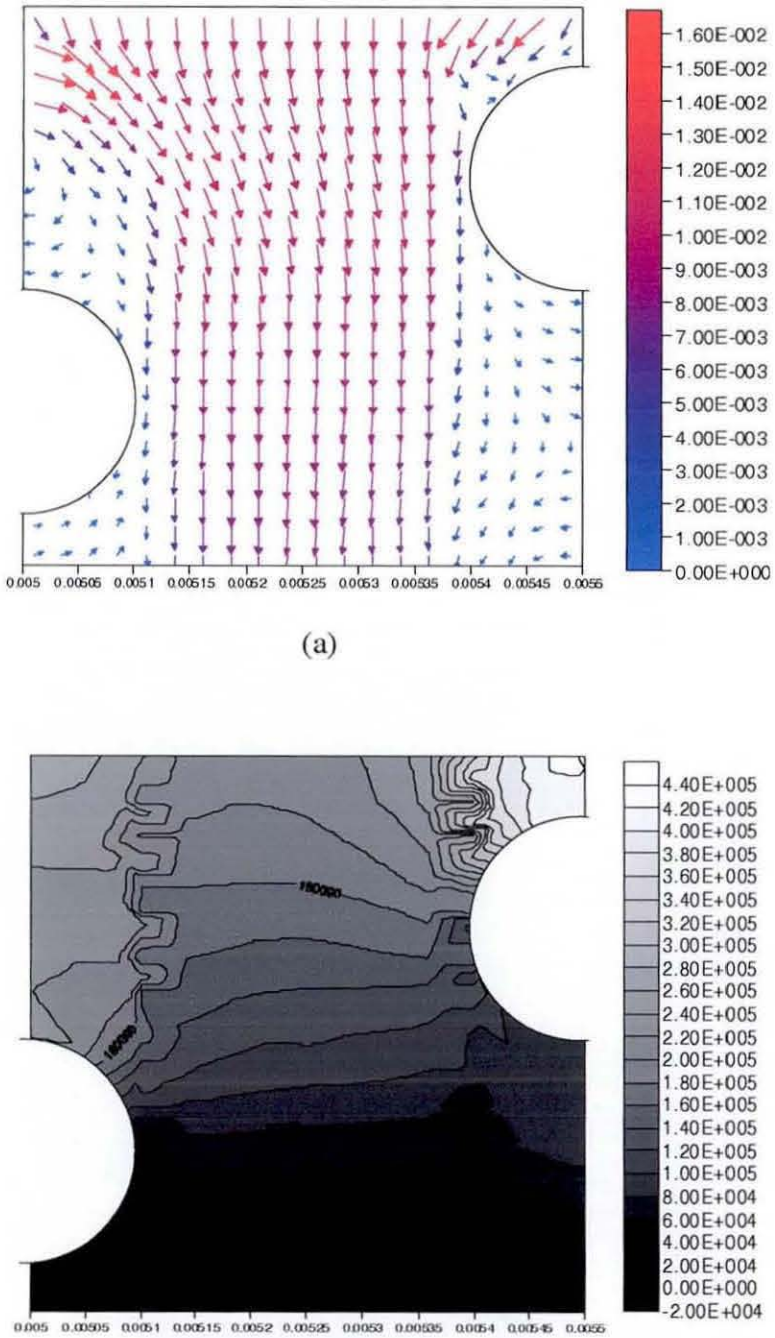
Figure 5.33 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 3 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

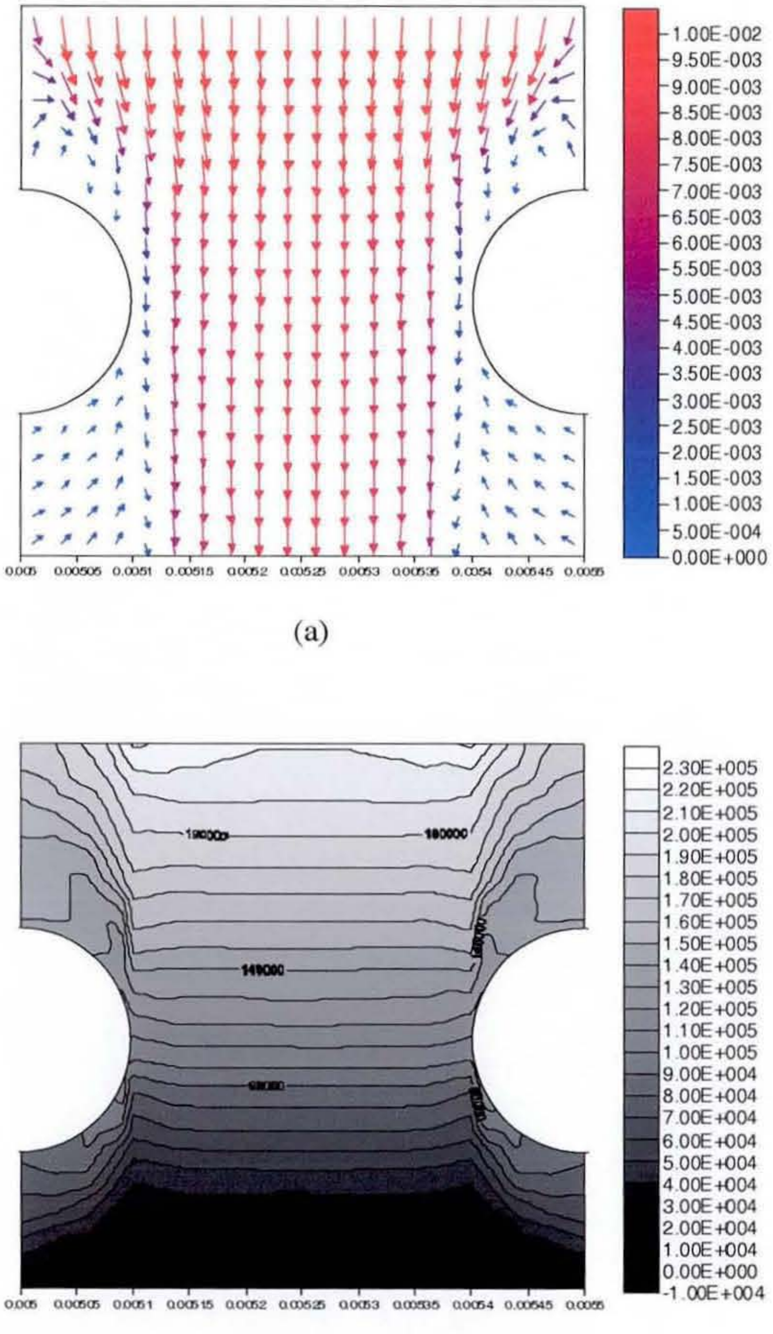
Figure 5.34 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 3 in the *y-z* plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

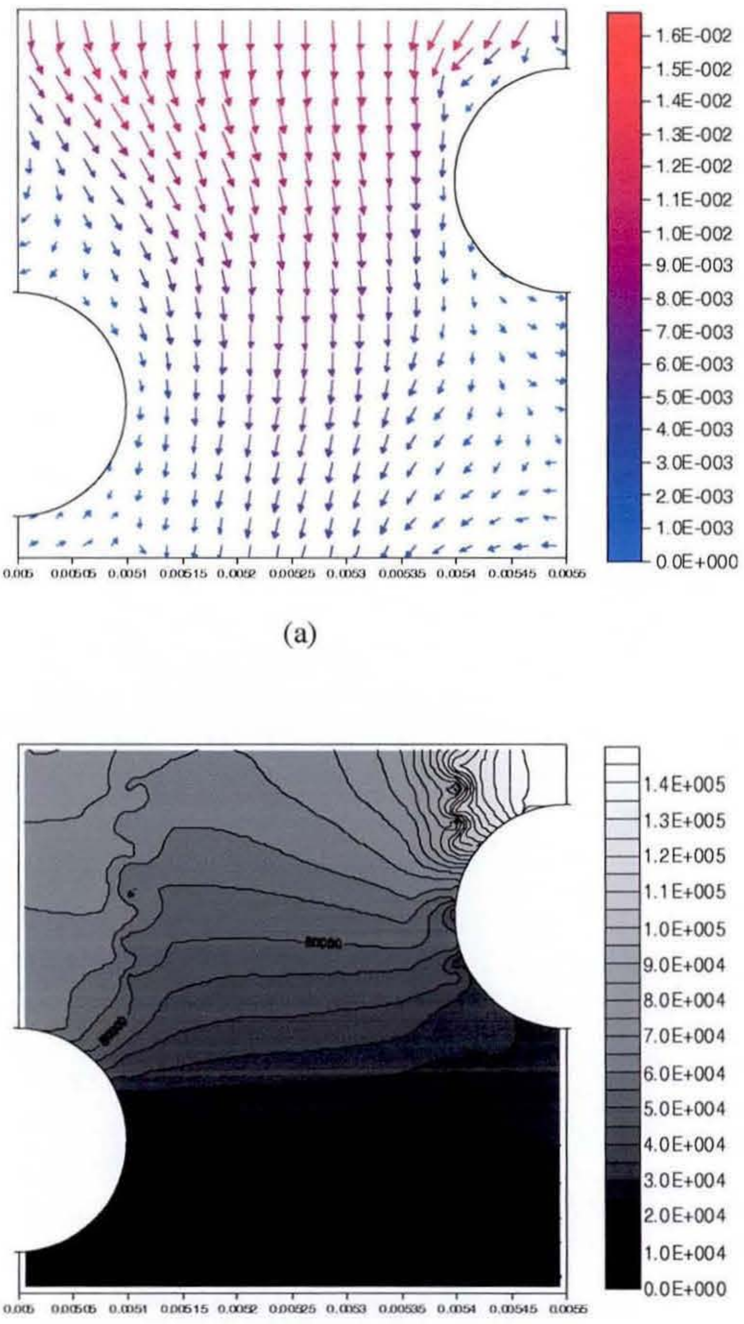
Figure 5.35 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 3 in the x-z plane at $y = l_w/2$ with inlet velocity v_z = -0.01ms^{-1}.



(a)

(b)

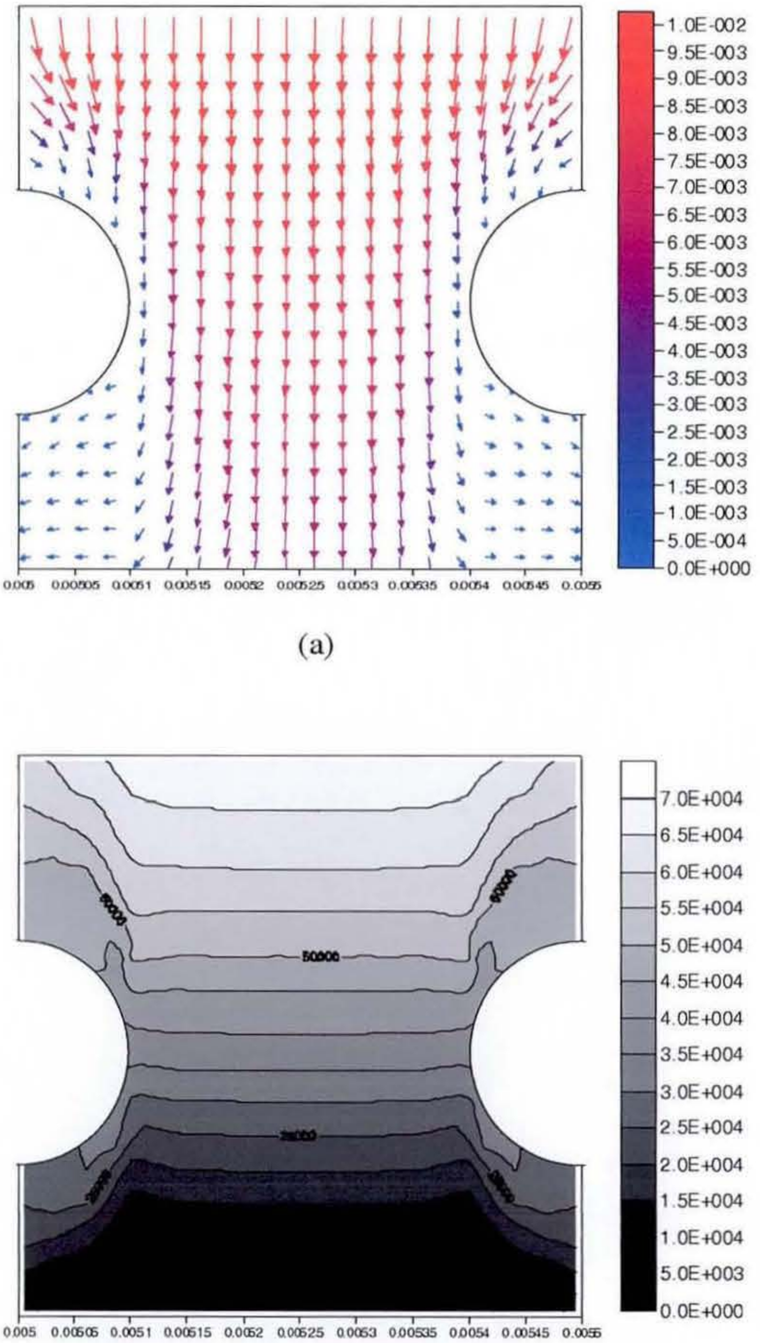
Figure 5.36 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 3 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.37 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 3 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.38 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 3 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

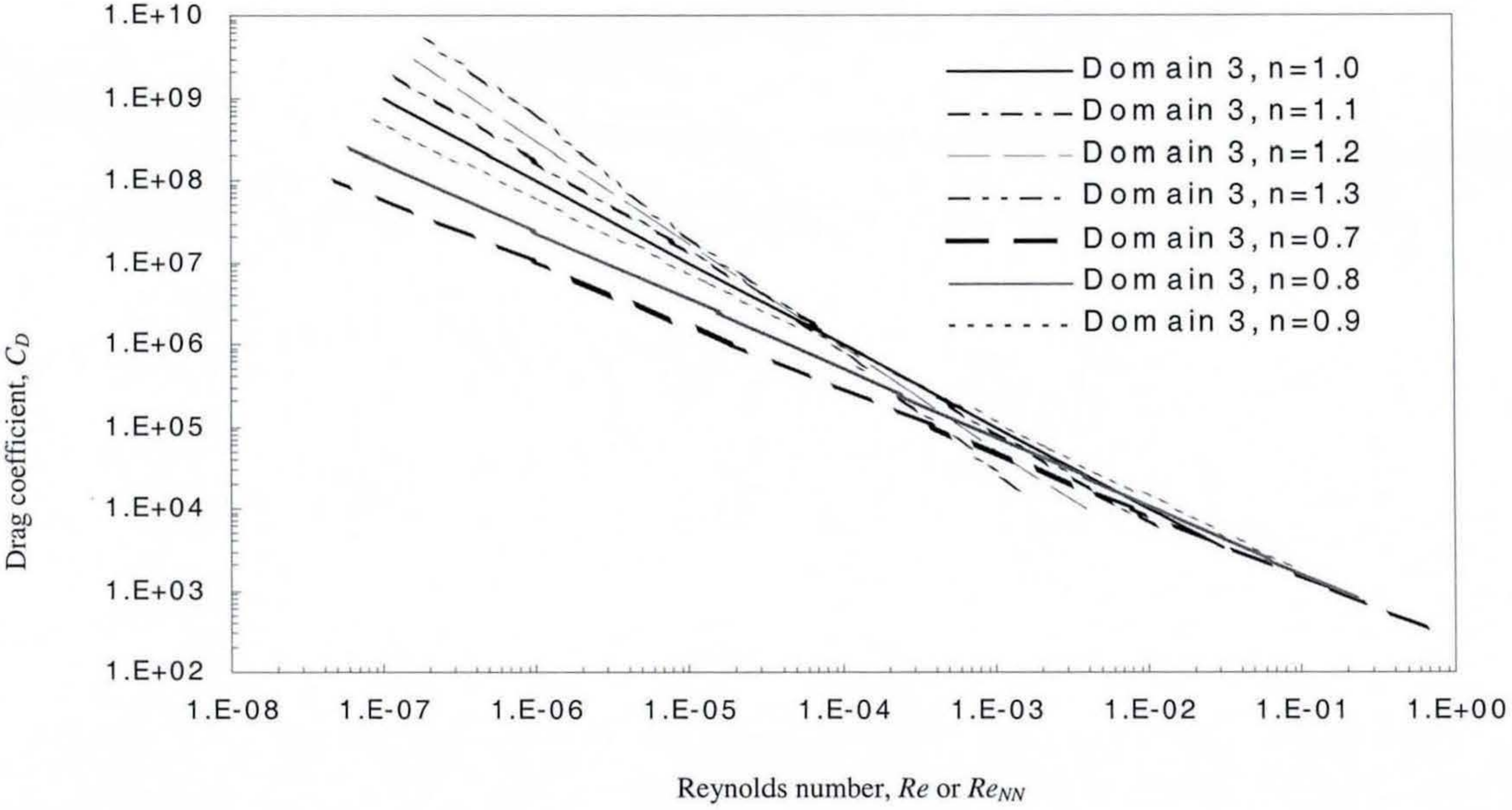
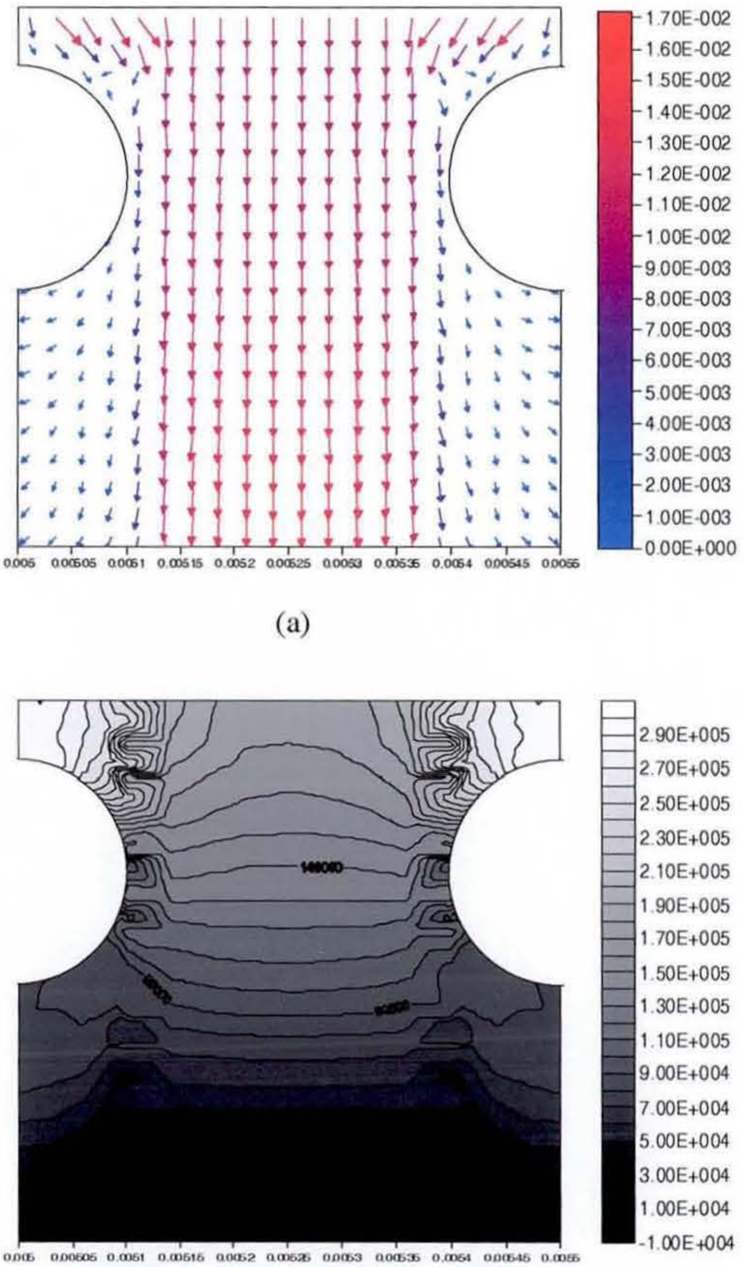


Figure 5.39 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 3.

5.7 Results from Domain 4: Pore Type 4 with Aperture/Diameter Ratio of 1.5

Typical velocity vectors for the flow of Newtonian fluid, shear thickening fluid and shear thinning fluids through domain 4 in the *x-z* plane at $y = l_w/2$ and *y-z* plane at $x = l_f/2$ are illustrated in Figures 5.40 - 5.45. Pore type 4 has the highest degree of symmetry where the *x-z* plane and *y-z* plane has full geometrical symmetry at every depth. The pattern of the fluid flow corresponding to the position of the wire mesh is consistent with the observations in domain 1, domain 2 and domain 3 where higher velocities were observed at the area immediately above the wire mesh and in the section between the wires. The velocity distribution and the pressure contours showed in Figures 5.40 - 5.45 are in perfect symmetry consistent with the symmetrical geometry.

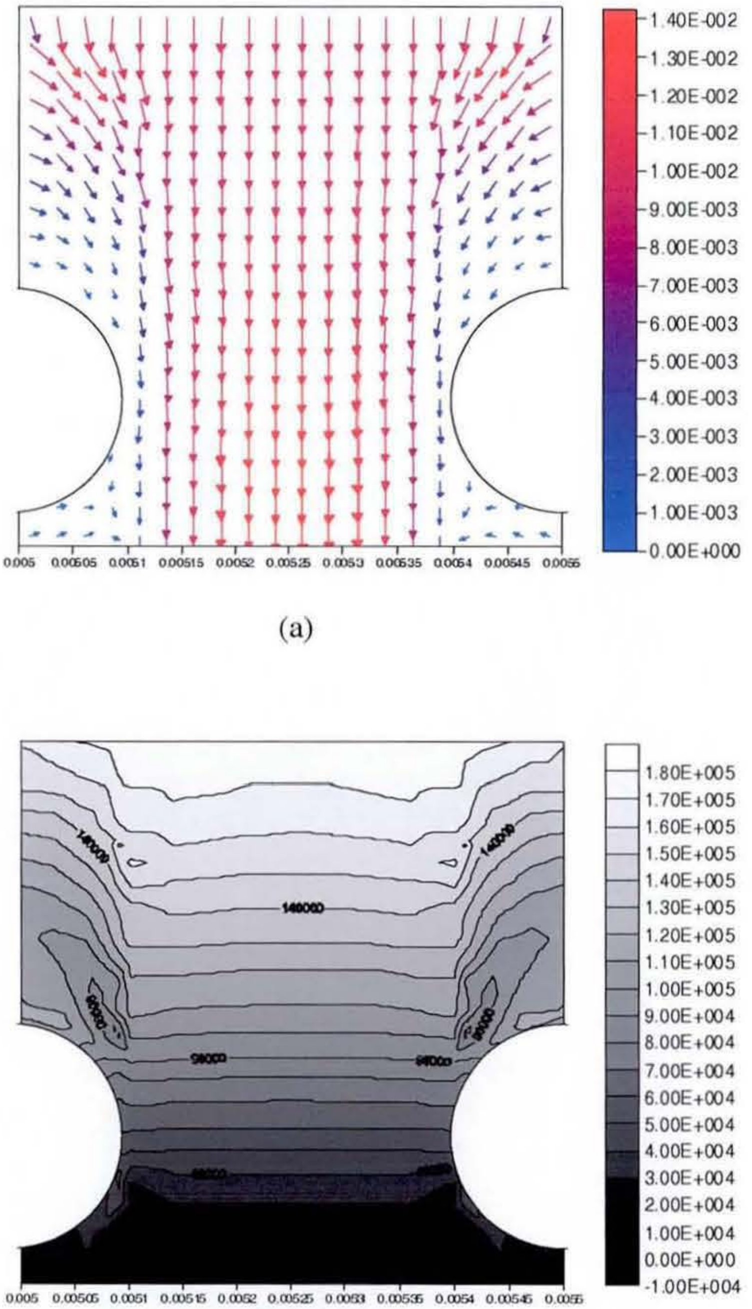
A corresponding rise in the simulated pressure drop value is observed when the power law index increases from 1.0 to 1.3, as observed in domain 1, 2 and 3. The pressure drop decreases when the power law index decreases from 1.0 to 0.7. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.46.



(a)

(b)

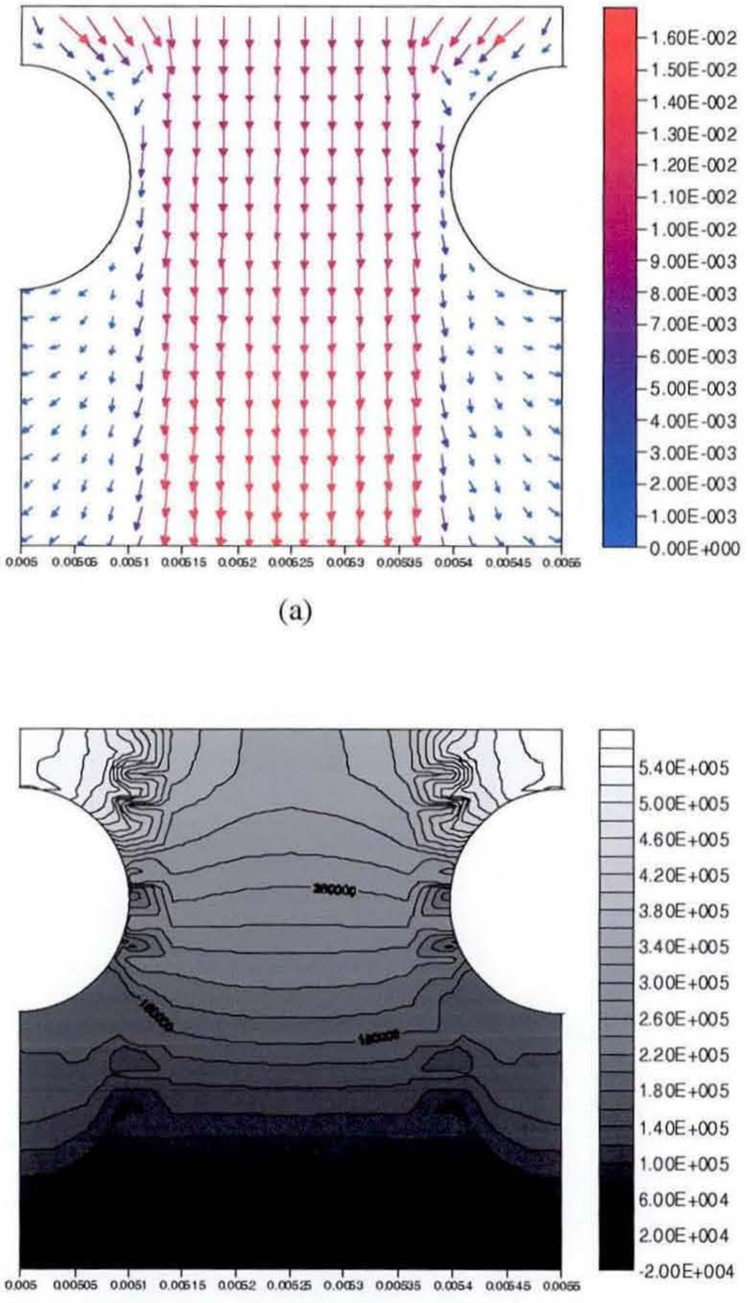
Figure 5.40 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 4 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

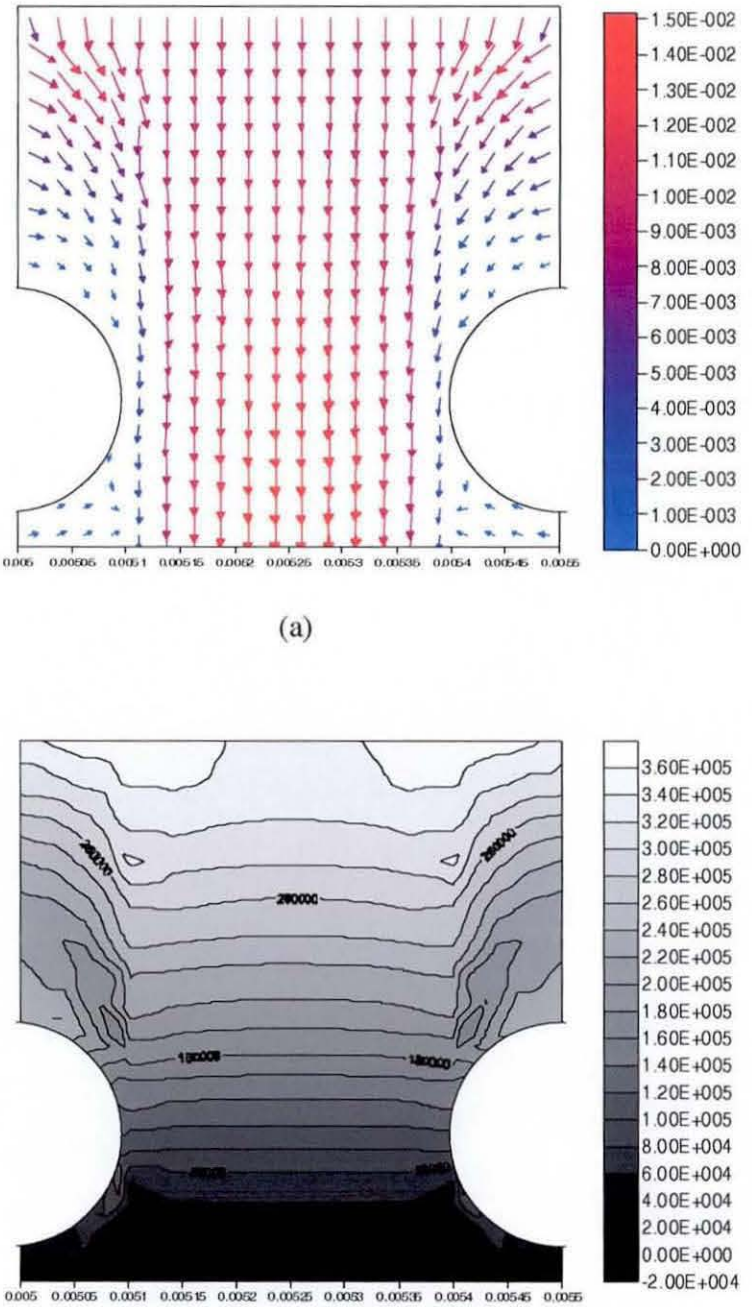
Figure 5.41 The (a) velocity vectors (b) pressure contour of Newtonian fluid flow through domain 4 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

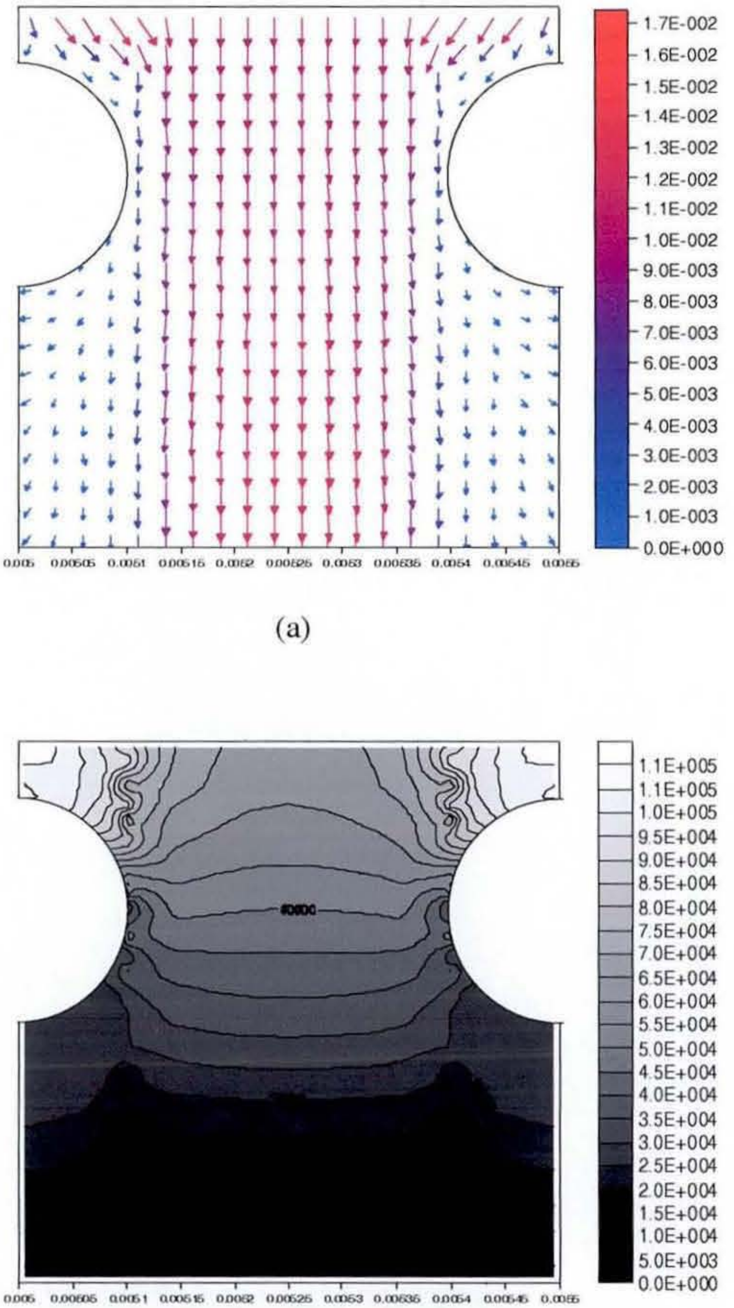
Figure 5.42 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 4 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

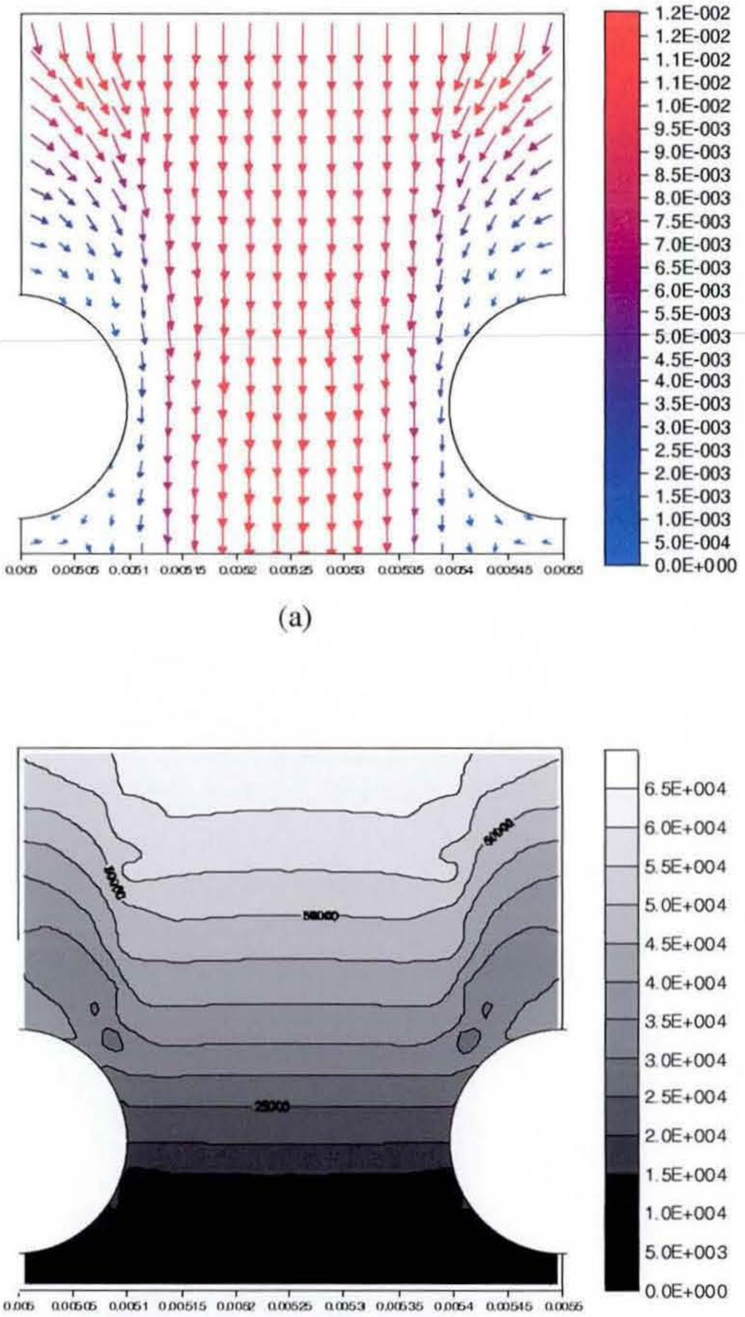
Figure 5.43 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 4 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.44 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 4 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.45 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 0.9$) flow through domain 4 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

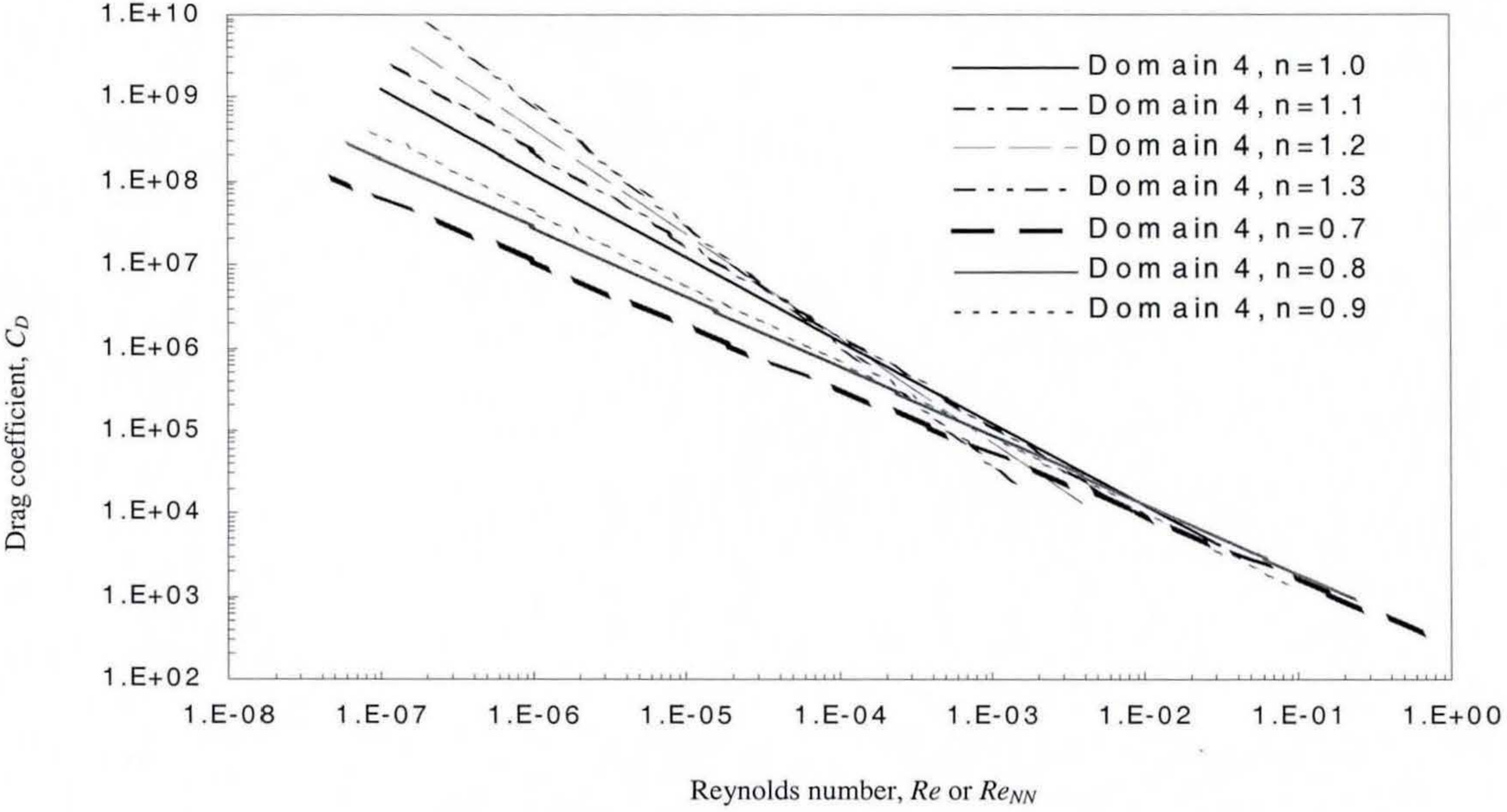
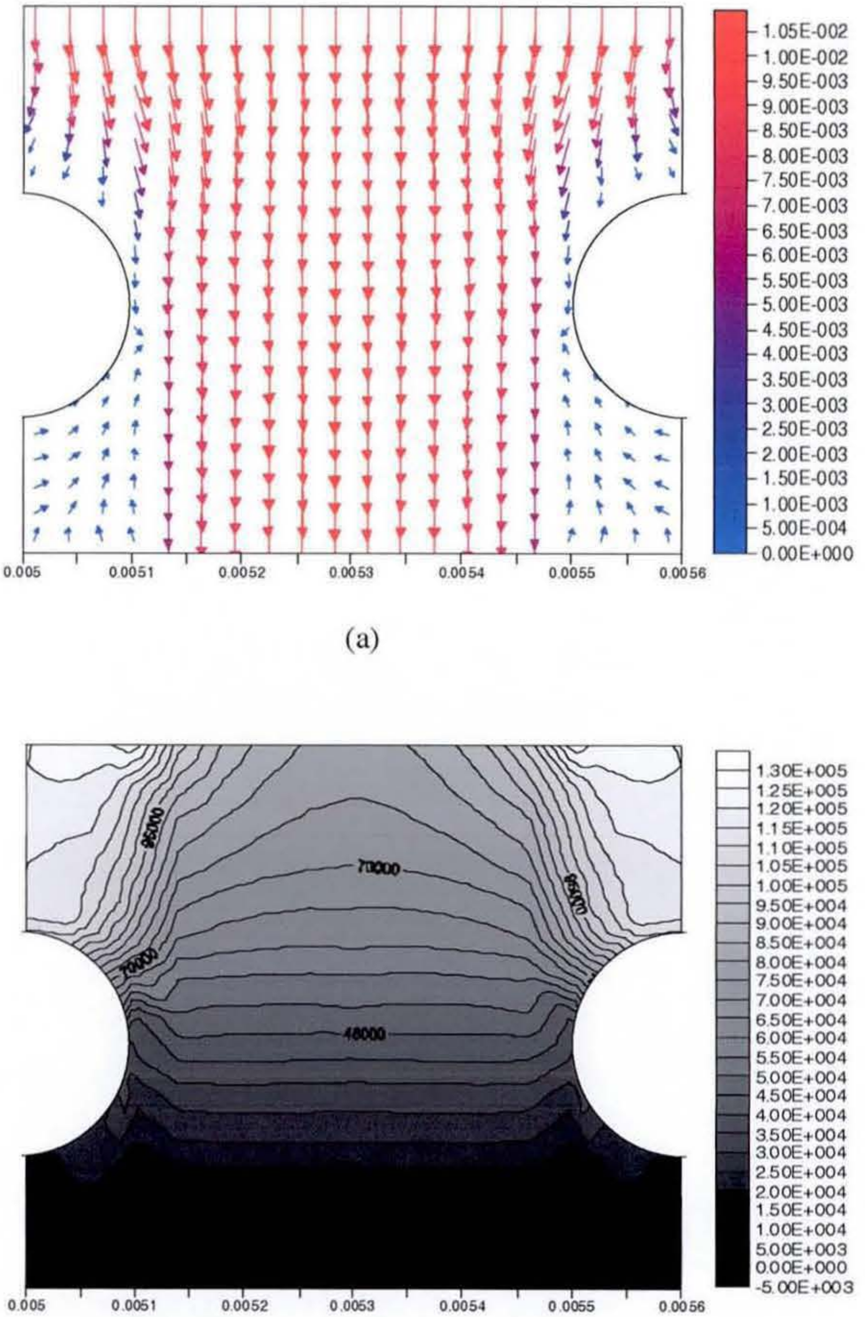


Figure 5.46 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 4.

5.8 Results from Domain 5: Pore Type 1 with Aperture/Diameter Ratio of 2.0

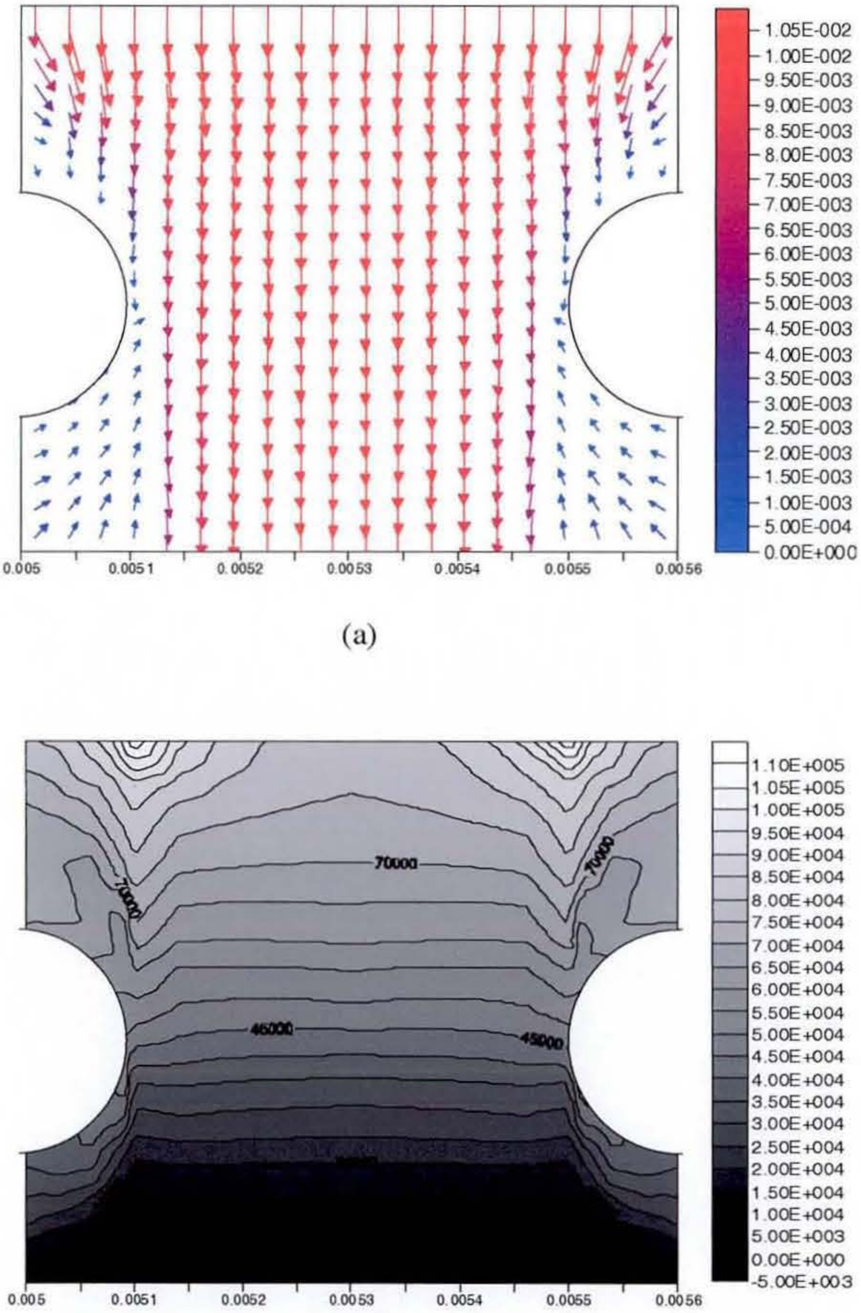
The typical velocity vectors for the flow of shear thickening fluid through domain 5 in the *x*-*z* plane at $y = l_w/2$ and *y*-*z* plane at $x = l_f/2$ are illustrated in Figures 5.47 and 5.48, respectively. The pattern of the fluid flow corresponding to the position of the wire mesh is observed to be similar to that of domain 1 where the flow is observed to swirl inwards towards the aperture on both *x*-*z* and *y*-*z* planes corresponding to the location of the wire mesh on both planes. Higher velocities are also observed at the area immediately above the wire mesh and in the section between the wires, compared to the velocity at the inlet and outlet.

Similar to the results reported earlier in domain 1, a corresponding rise in the simulated pressure drop value is observed in domain 5 when the power law index increases from 1.0 to 1.3. The pressure drop decreases in similar fashion when the power law index decreases from 1.0 to 0.7. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.49.



(a)

Figure 5.47 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 5 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.48 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 5 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

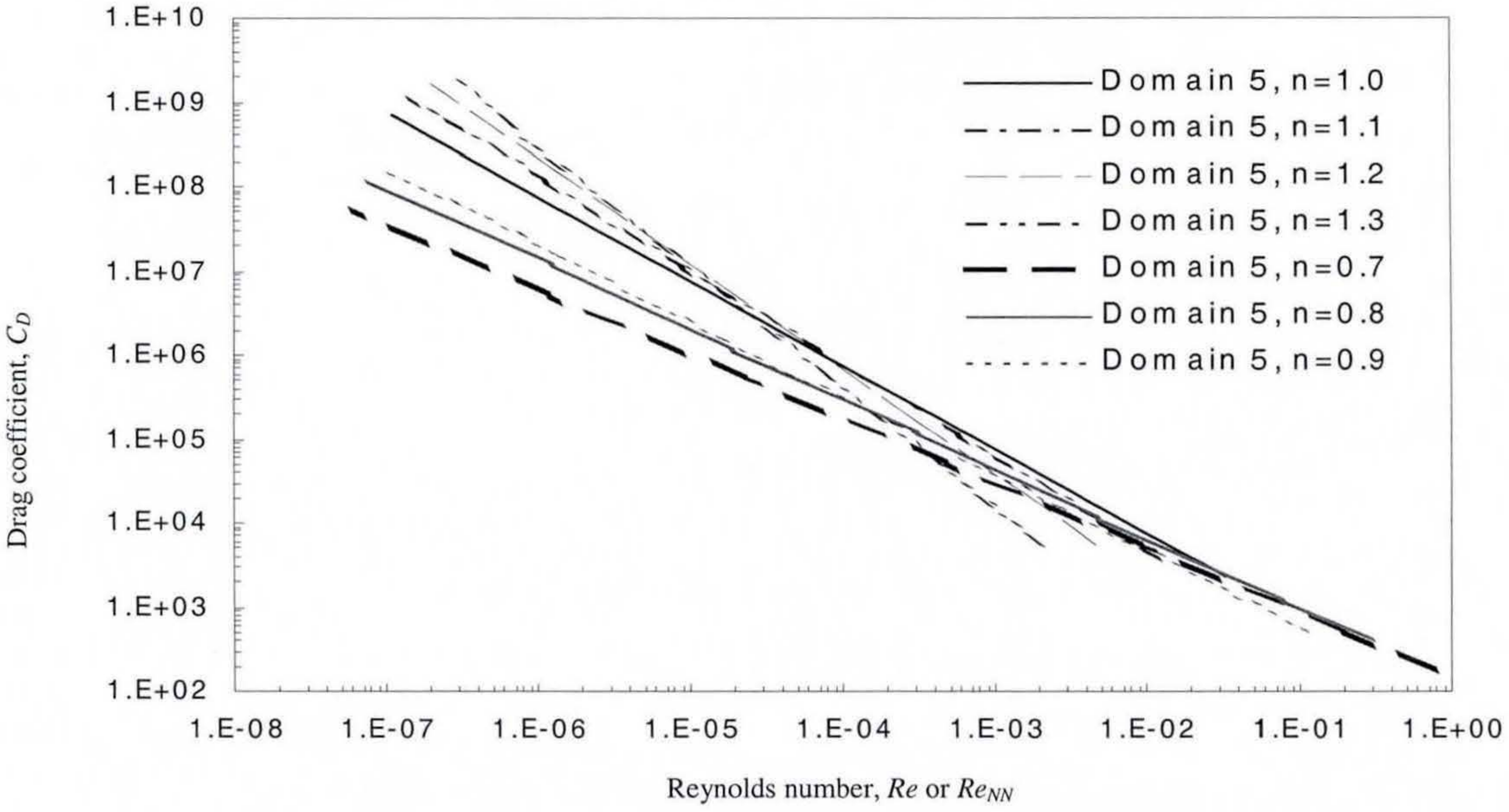


Figure 5.49 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 5.

Compared to results presented for domain 1, the pressure drop over the interstices in domain 5 were observed to be lower generally. This is due to the higher cross-sectional area across domain 5 for the fluid to flow through compared to the area available in domain 1. This is consistent with the phenomena observed experimentally by Rushton (1969). The difference in the pressure drop for domain 1 and domain 5 is observed to be in the region of 2.9% to 21%, with the difference becoming more pronounced as power law index n increases. The C_D versus Re_{NN} plot for domain 1 and domain 5 for Newtonian fluid is given in Figure 5.50. It should be remembered that the C_D versus Re_{NN} graph is plotted with the porosity ε taken into consideration. As such, the graph has to be interpreted in relation to the factor ε.

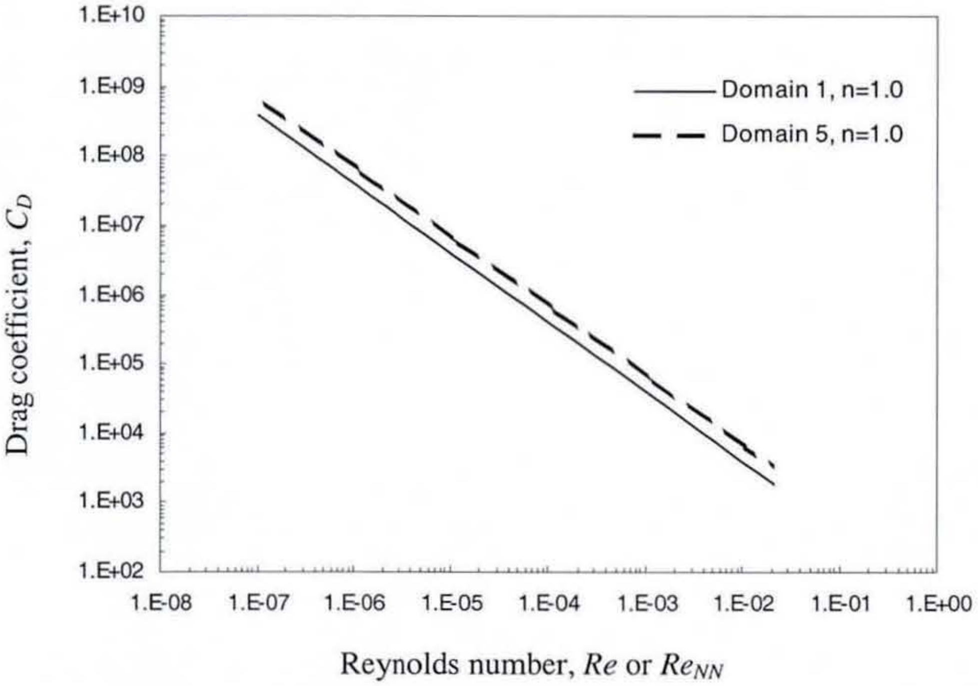


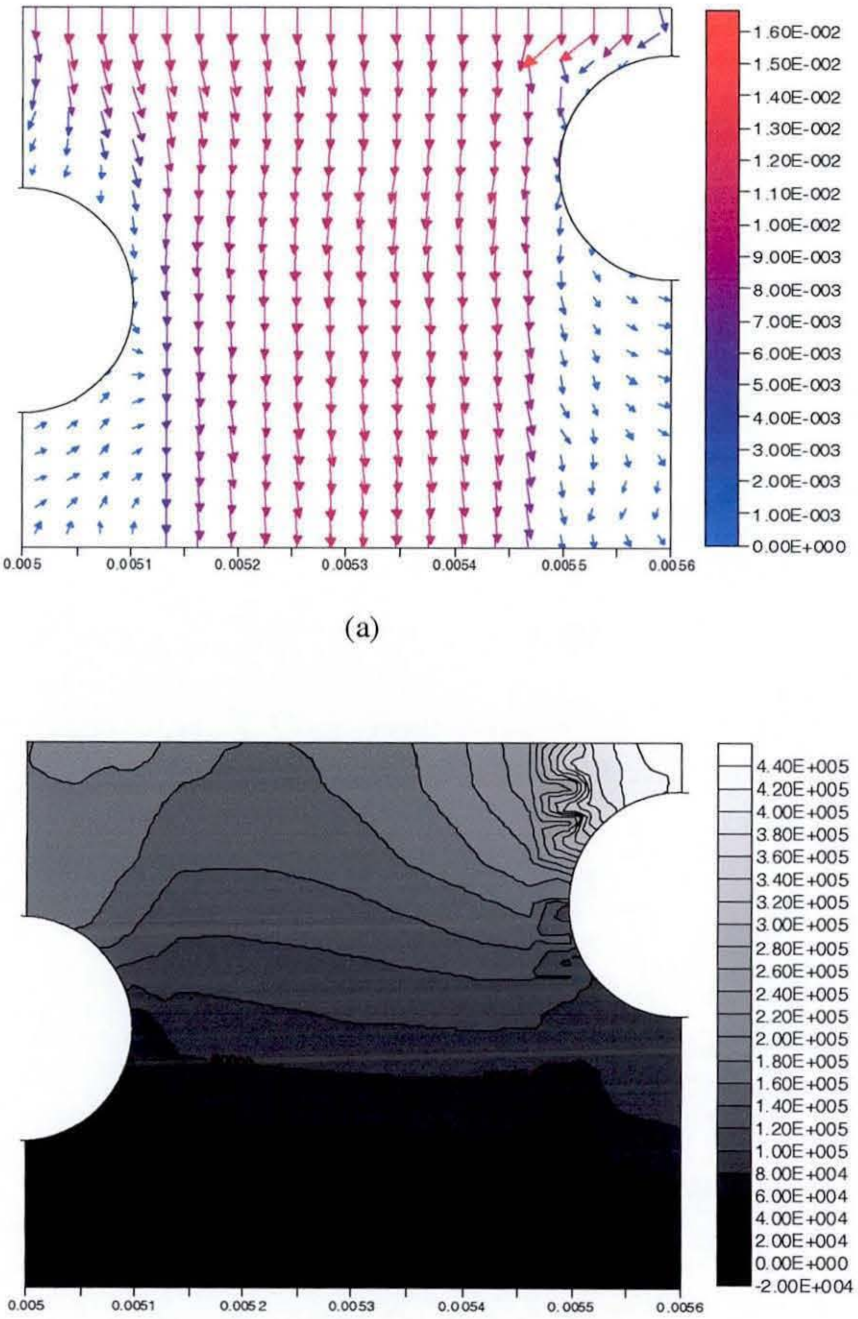
Figure 5.50 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domain 1 and domain 5.

5.9 Results from Domain 6: Pore Type 2 with Aperture/Diameter Ratio of 2.0

The typical velocity vectors for the flow of shear thickening fluid through domain 6 in the *x-z* plane at $y = l_w/2$ and *y-z* plane at $x = l_f/2$ are illustrated in Figures 5.51 and 5.52, respectively. The pattern of the fluid flow corresponding to the position of the wire mesh is observed to be similar to that of domain 2 where the pattern of the flow reflected the absence of symmetrical pattern in both planes. The velocity above the right wire mesh in the *x-z* plane plots is higher than that above the left wire mesh, similar to that observed in domain 2.

Similar to the results reported earlier in domain 2, the simulated pressure drop value in domain 5 is observed to rise when the power law index increases from 1.0 to 1.3. The pressure drop decreases corresponding to a decrease of the power law index. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.53.

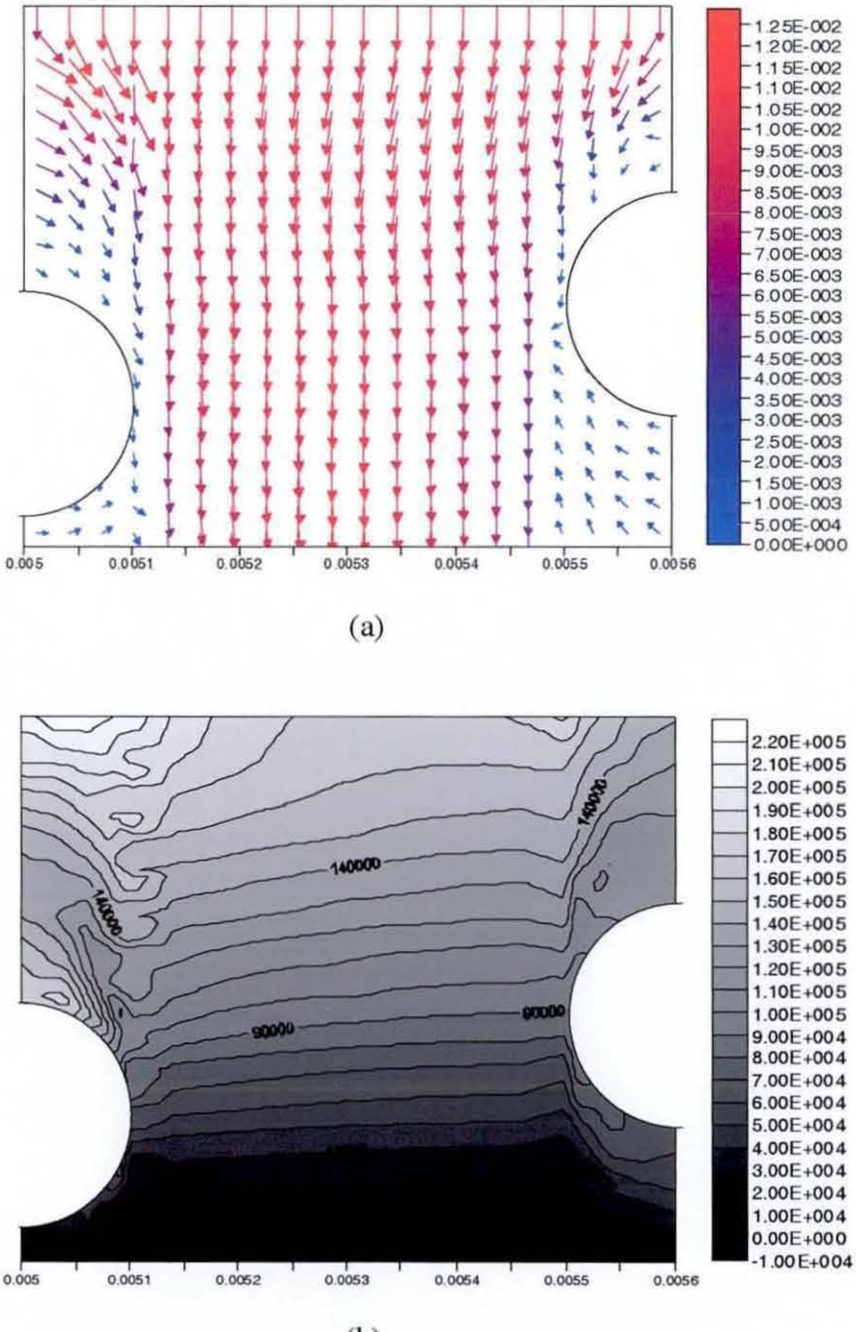
Again, lower pressure drop over the interstices in domain 6 was observed compared to domain 2. The difference in the pressure drop for domain 2 and domain 6 is less dramatic compared to the differences observed for domain 1 and domain 5. A quantitative examination showed the maximum pressure drop difference is 4% and decreases as power law index n decreases. The C_D versus Re_{NN} plot for domain 2 and domain 6 for Newtonian fluid is given in Figure 5.54.



(a)

(b)

Figure 5.51 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 6 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.



(a)

(b)

Figure 5.52 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 1.1$) flow through domain 6 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

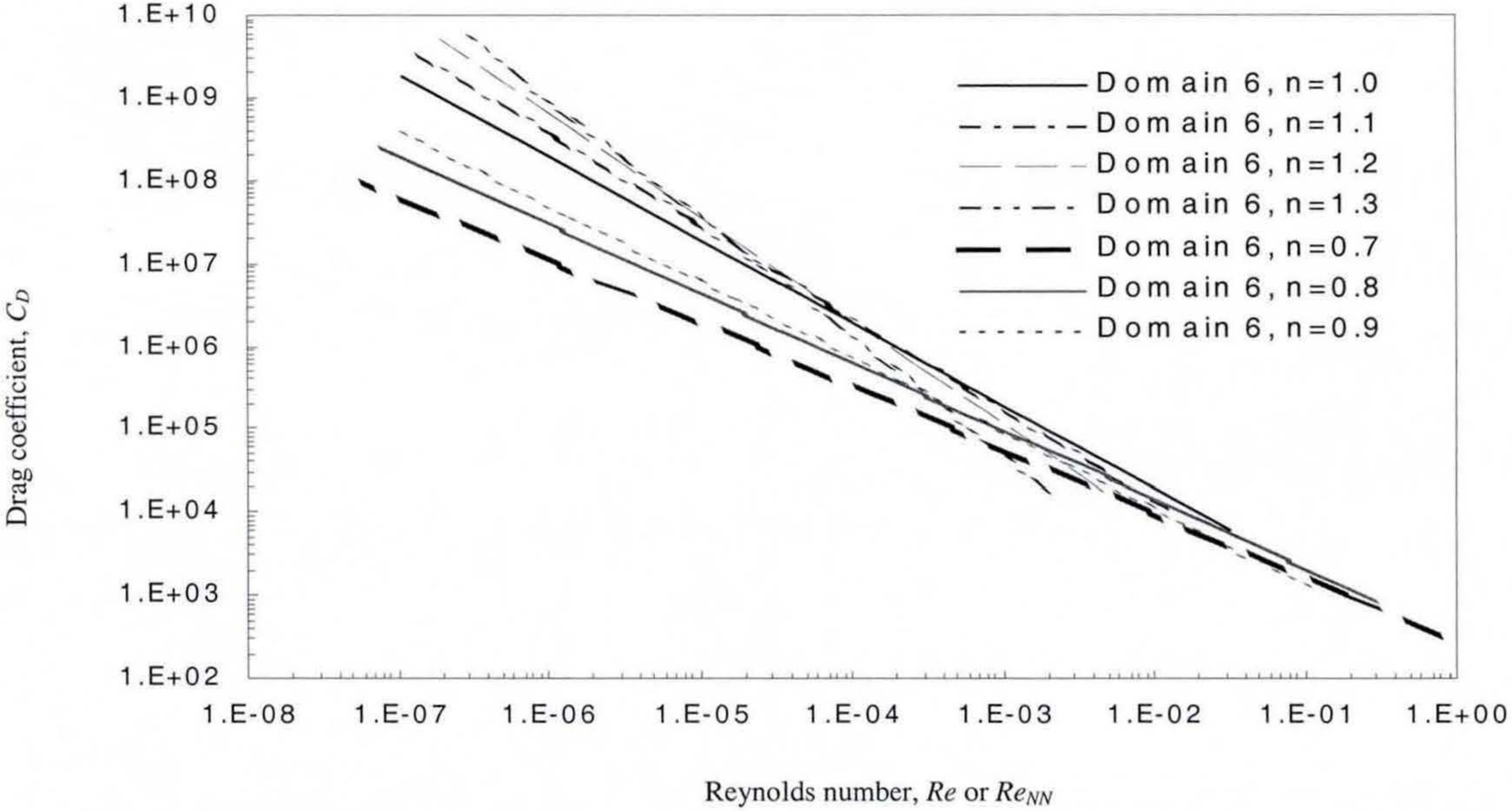


Figure 5.53 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 6.

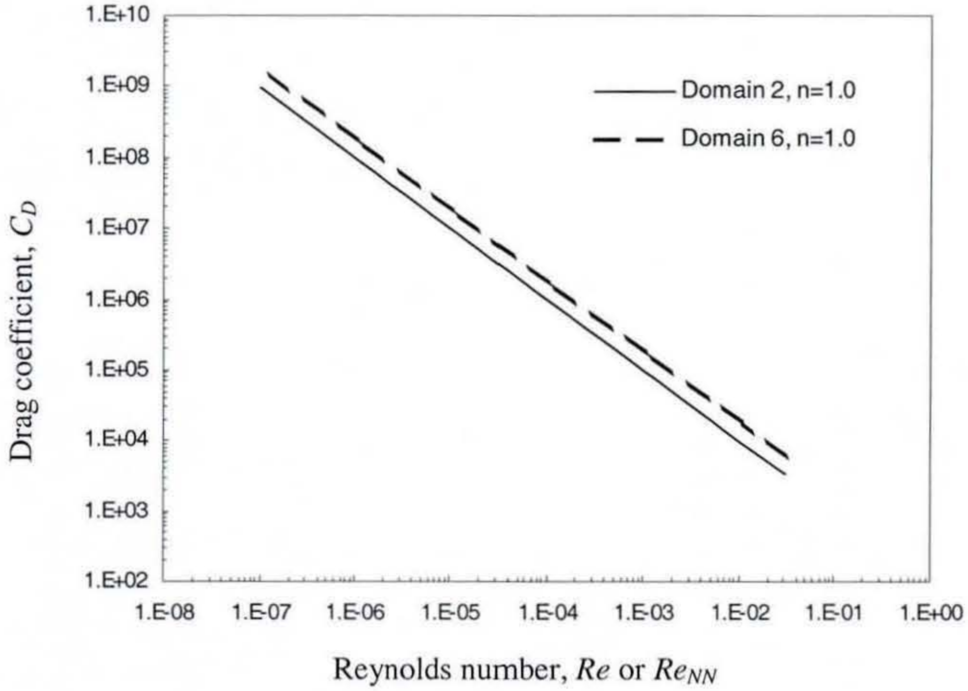


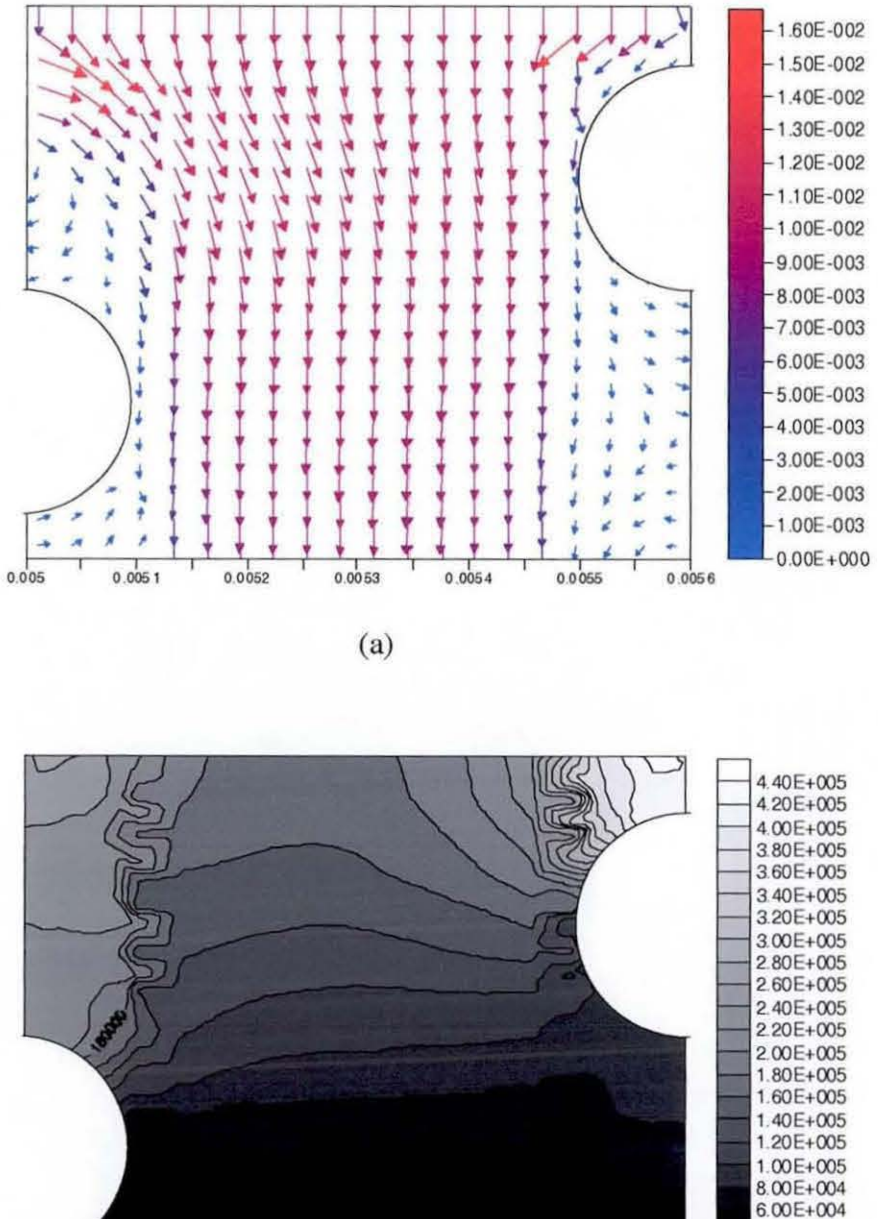
Figure 5.54 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domain 2 and domain 6.

5.10 Results from Domain 7: Pore Type 3 with Aperture/Diameter Ratio of 2.0

The typical velocity vectors for the flow of shear thickening fluid through domain 7 in the x-z plane at $y = l_w/2$ and y-z plane at $x = l_f/2$ are illustrated in Figures 5.55 and 5.56, respectively. The pattern of the fluid flow in domain 7 is similar to that of domain 3 where a symmetrical flow can be observed in the y-z plane but not in the x-z plane. The pattern of the fluid flow in the x-z plane corresponding to the position of the wire mesh is similar to that observed in domain 6 while flow pattern in the y-z plane is similar to that shown for domain 5.

Similar to domains 5 and 6, as the power law index increases from 1.0 to 1.3, there is a corresponding rise in the simulated pressure drop value. The pressure drop decreases when the power law index decreases from 1.0 to 0.7. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.57.

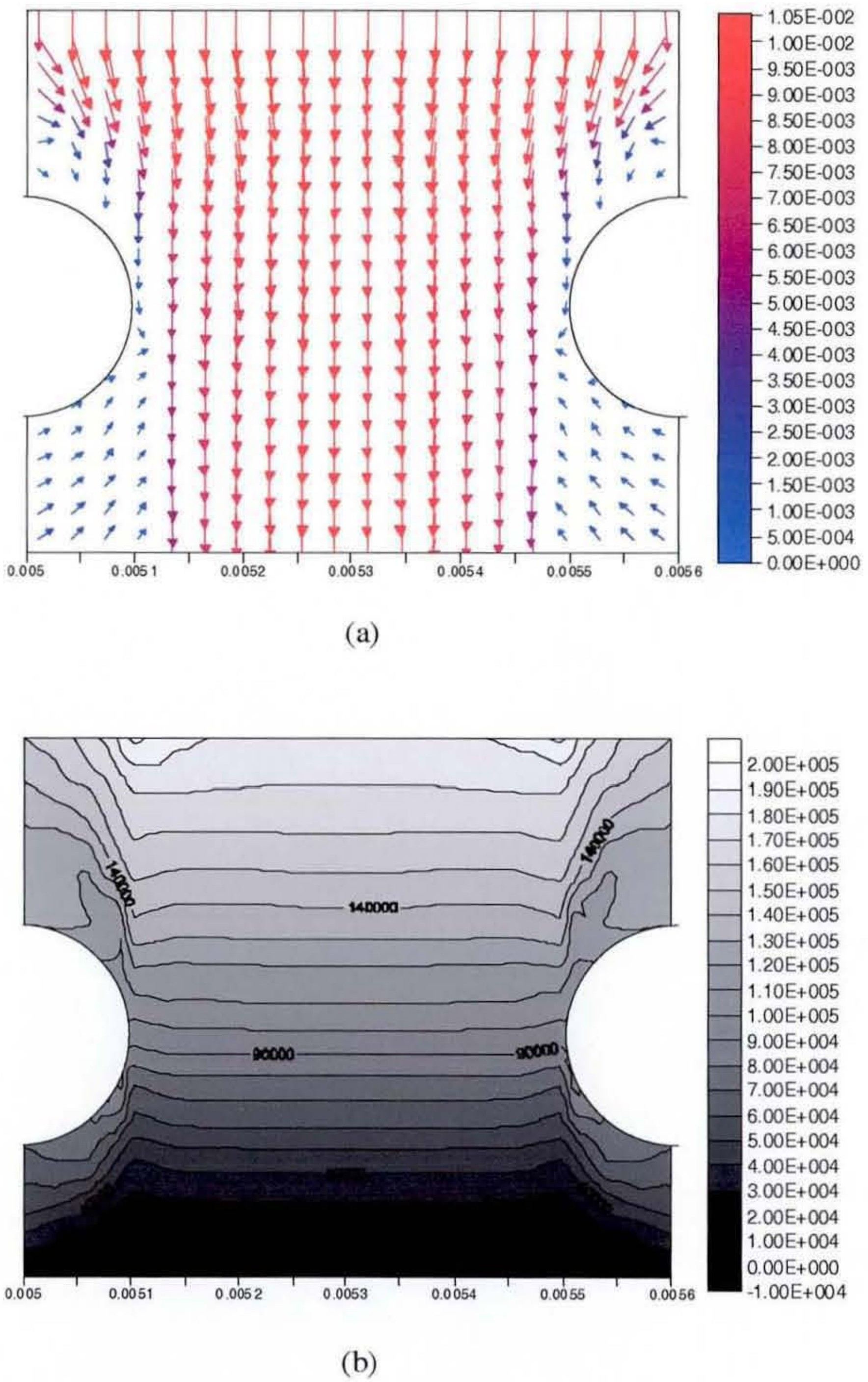
In a similar pattern observed for domain 1 and 5, and domain 2 and 6, lower pressure drop over the interstices in domain 7 was observed compared to domain 3. The maximum pressure drop difference for domain 3 and domain 7 is 4.9% and decreases as power law index n decreases. The C_D versus Re_{NN} plot for domain 3 and domain 7 for Newtonian fluid is given in Figure 5.58.



(a)

(b)

Figure 5.55 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 7 in the x-z plane at $y = l_w/2$ with inlet velocity v_z = -0.01ms^{-1}.



(a)

(b)

Figure 5.56 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid ($n = 1.1$) flow through domain 7 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

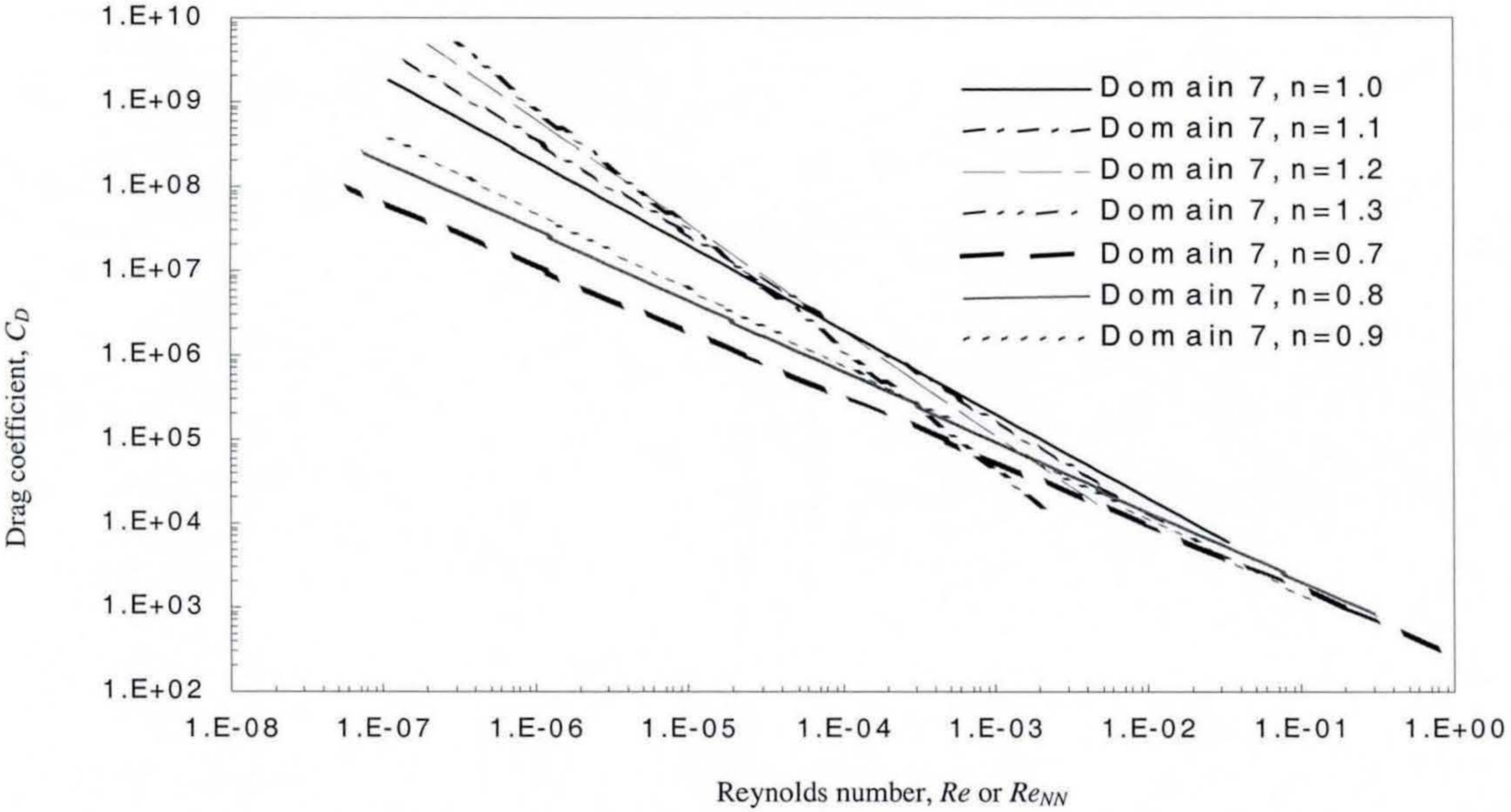


Figure 5.57 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 7.

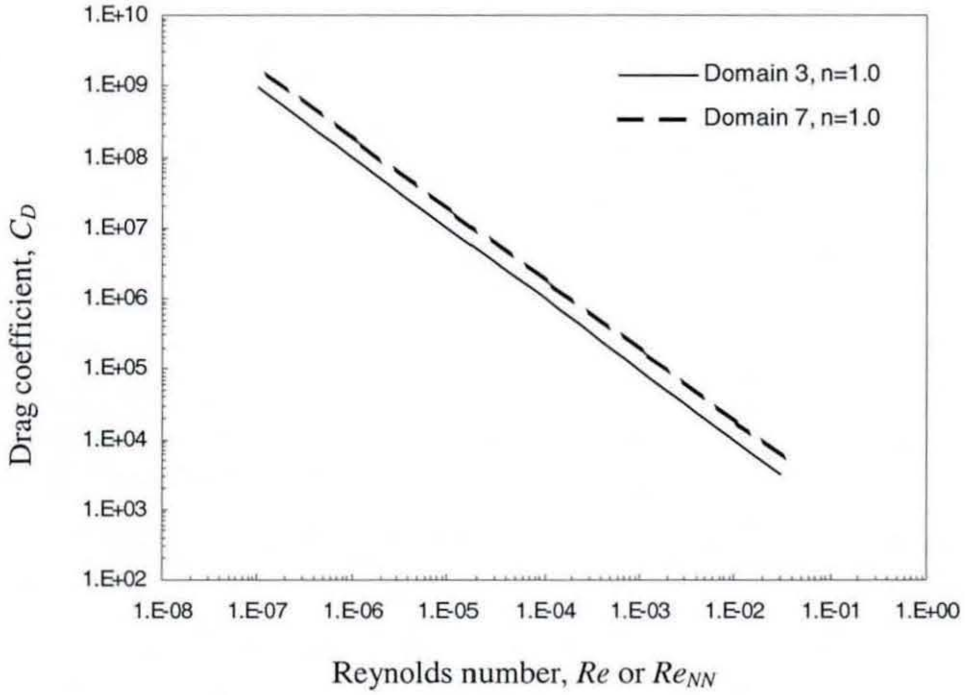


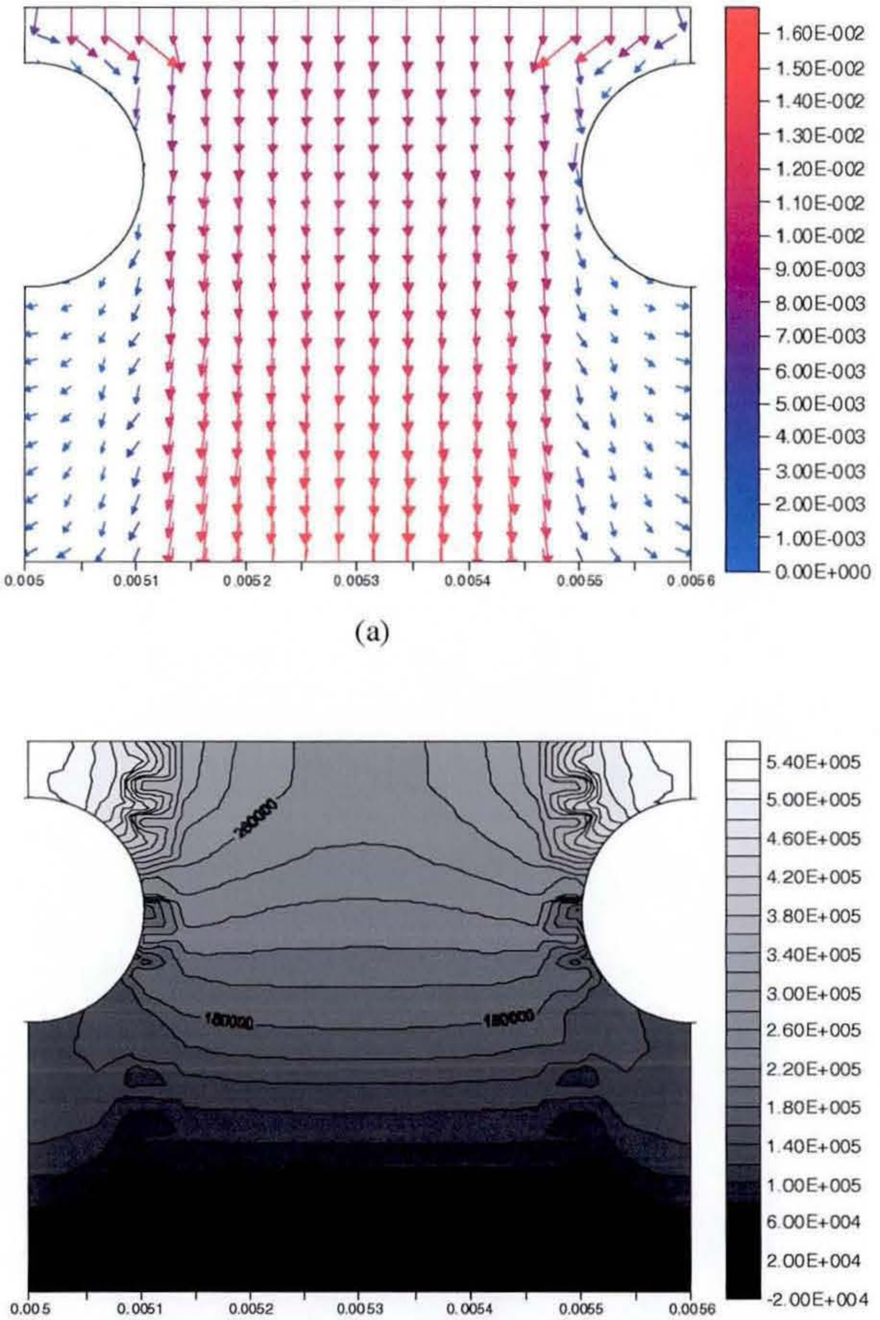
Figure 5.58 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domain 3 and domain 7.

5.11 Results from Domain 8: Pore Type 4 with Aperture/Diameter Ratio of 2.0

The typical velocity vectors for the flow of shear thickening fluid through domain 8 in the x-z plane at $y = l_w/2$ and y-z plane at $x = l_f/2$ are illustrated in Figures 5.59 and 5.60, respectively. The pattern of the fluid flow illustrated by velocity distribution and pressure contours reflected the symmetrical geometry at both x-z plane and y-z planes. This is consistent with the observation in results presented for domain 4. Similar to domain 4, the flow is observed to swirl inwards towards the aperture on both x-z and y-z planes corresponding to the location of the wire mesh on both planes. Higher velocities are also observed at the area immediately above the wire mesh and in the section between the wires, compared to the velocity at the inlet and outlet.

A corresponding rise in the simulated pressure drop value is observed when the power law index increases from 1.0 to 1.3, as observed in domain 5, 6, and 7. The pressure drop decreases as the power law index decreases. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.61.

Compared to domain 4, domain 8 was observed to give lower pressure drop across the interstices. The maximum pressure drop difference for domain 4 and domain 8 is 5.6% and decreases as power law index n decreases. The lower pressure drop has been consistently observed in domains with similar weave pattern but with higher aperture to diameter ratios as discussed earlier for domains 1 and 5, domains 2 and 6 and domains 3 and 7. The C_D versus Re_{NN} plot for domain 4 and domain 8 for Newtonian fluid is given in Figure 5.62.



(a)

(b)

Figure 5.59 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 8 in the x-z plane at $y = l_w/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

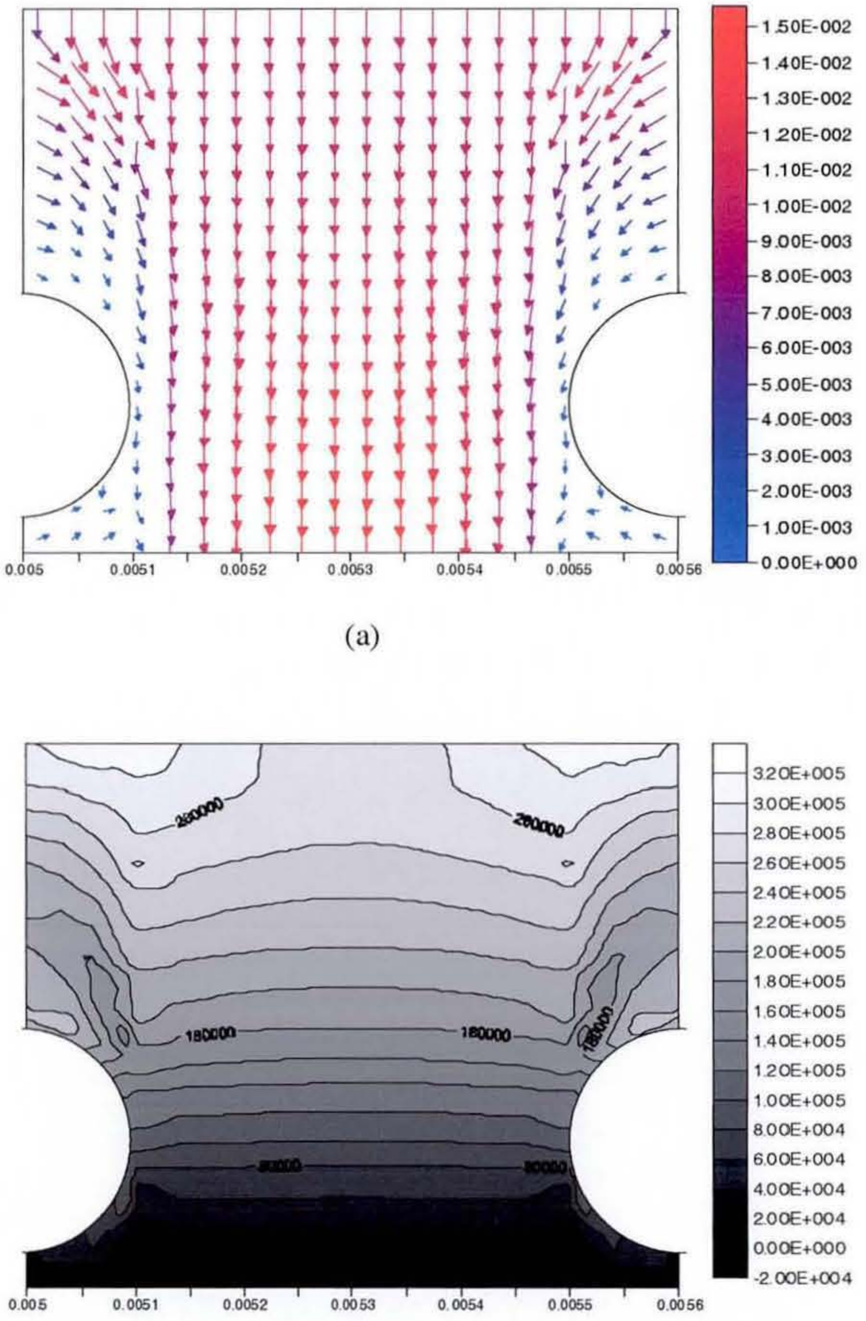


Figure 5.60 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 8 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

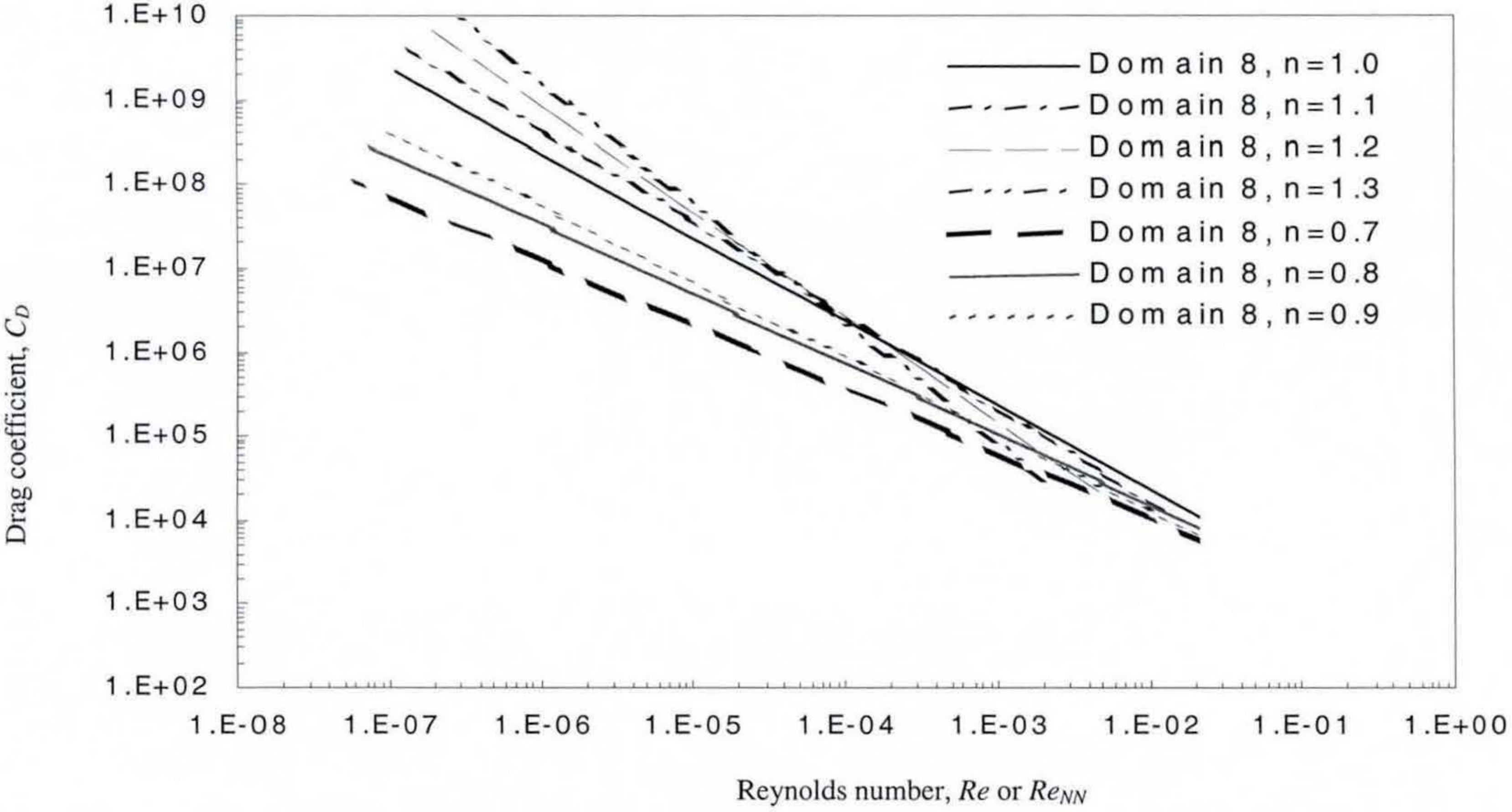


Figure 5.61 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 8.

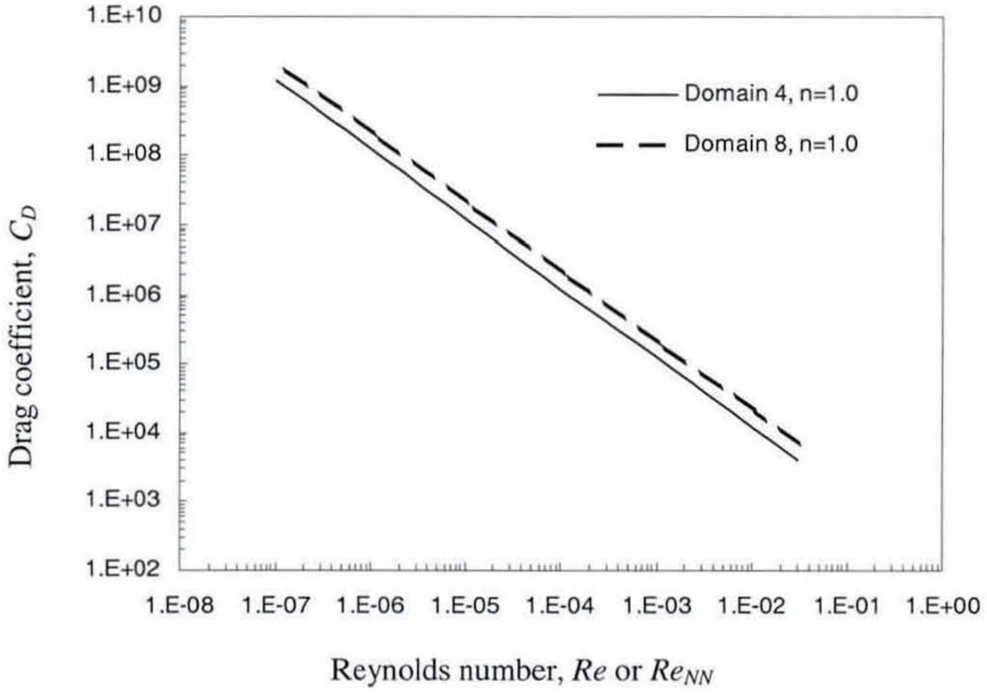


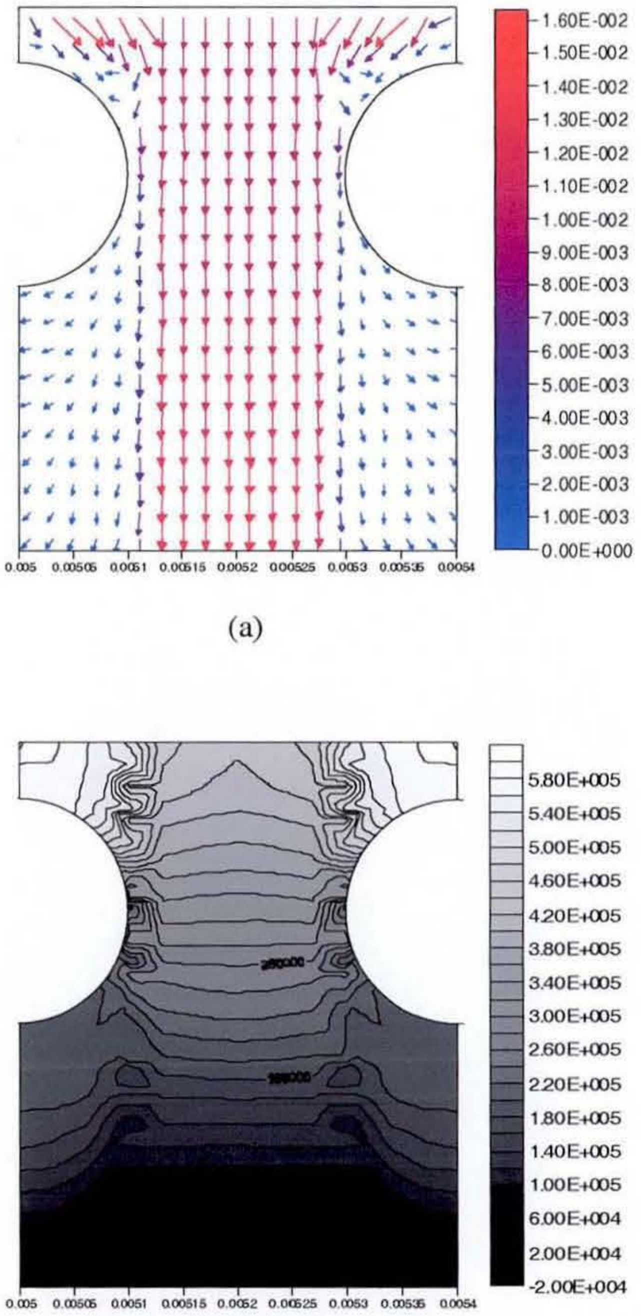
Figure 5.62 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domain 4 and domain 8.

5.12 Results from Domain 9: Pore Type 4 with Aperture/Diameter Ratio of 1.0

The typical velocity vectors for the flow of shear thickening fluid through domain 9 in the *x*-*z* plane at $y = l_w/2$ and *y*-*z* plane at $x = l_f/2$ are illustrated in Figures 5.63 and 5.64. The pattern of the fluid flow illustrated by velocity distribution and pressure contours reflected the symmetrical geometry at both *x*-*z* plane and *y*-*z* planes, consistent with the observation in results presented for domains 4 and 8.

A corresponding rise in the simulated pressure drop value is also observed when the power law index increases from 1.0 to 1.3, as observed in domains 4 and 8. Similarly, the pressure drop decreases when the power law index decreases. The drag coefficient versus Reynolds number plot corresponding to various power law indices are presented in Figure 5.65.

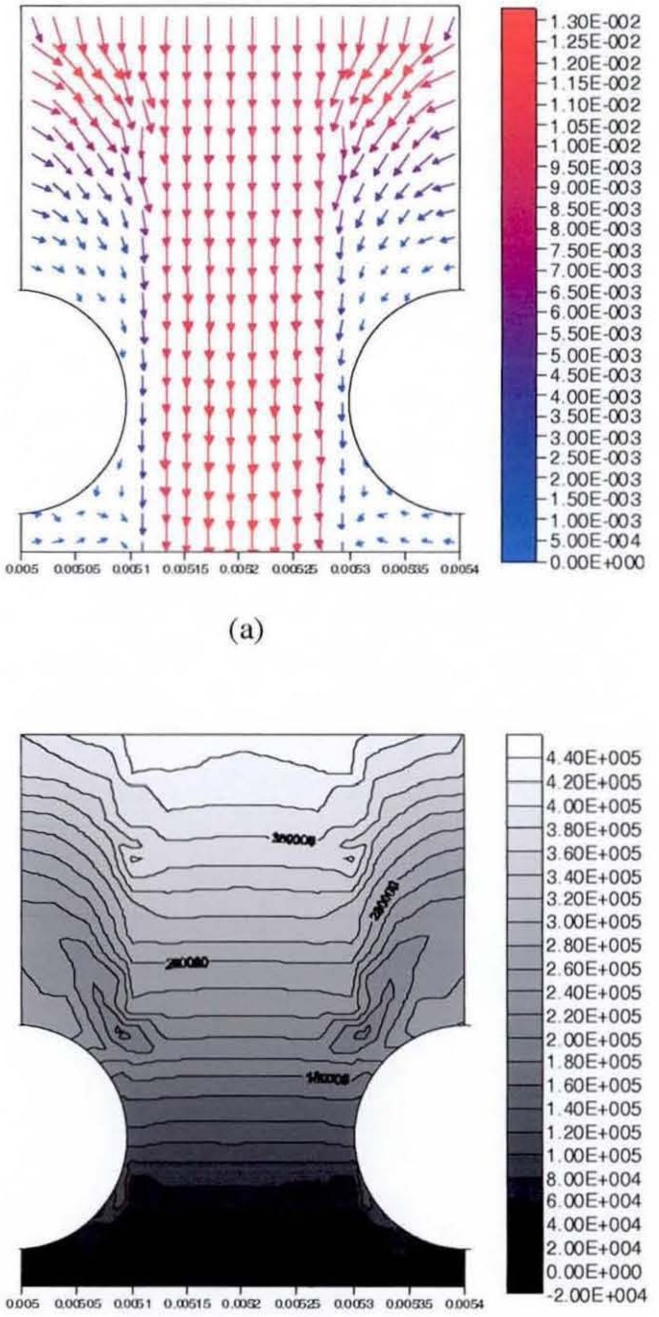
Compared to domains 4 and 8, domain 9 was observed to give higher pressure drop across the interstices. The maximum difference in pressure drop is 11% higher to that of domain 4 and it decreases as power law index *n* decreases. Comparing domains 4, 8 and 9, domain 9 was observed to give the highest pressure drop, followed by domain 4 and 8. This observation is consistent with the cross-sectional area available for flow in which higher aperture to diameter ratio rendered in higher area for flow, thus giving lower flow resistance resulting in lower pressure drop. The C_D versus Re_{NN} plot for domains 4, 8 and 9 for Newtonian fluid is given in Figure 5.66.



(a)

(b)

Figure 5.63 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 9 in the x-z plane at $y = l_w/2$ with inlet velocity v_z = -0.01ms^{-1}.



(a)

(b)

Figure 5.64 The (a) velocity vectors (b) pressure contour of non-Newtonian fluid (n = 1.1) flow through domain 9 in the y-z plane at $x = l_f/2$ with inlet velocity $v_z = -0.01\text{ms}^{-1}$.

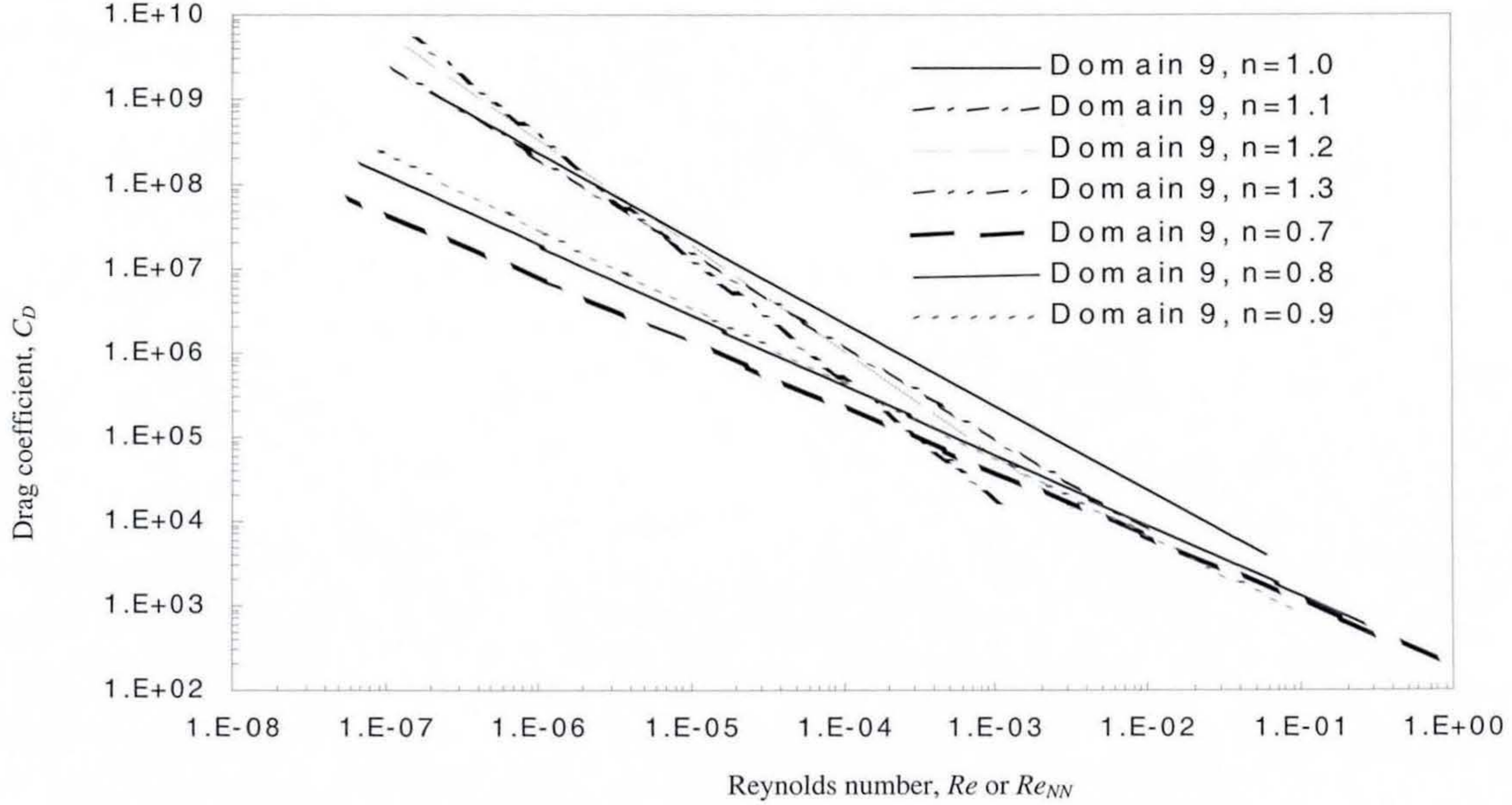


Figure 5.65 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian fluids flow through domain 9.

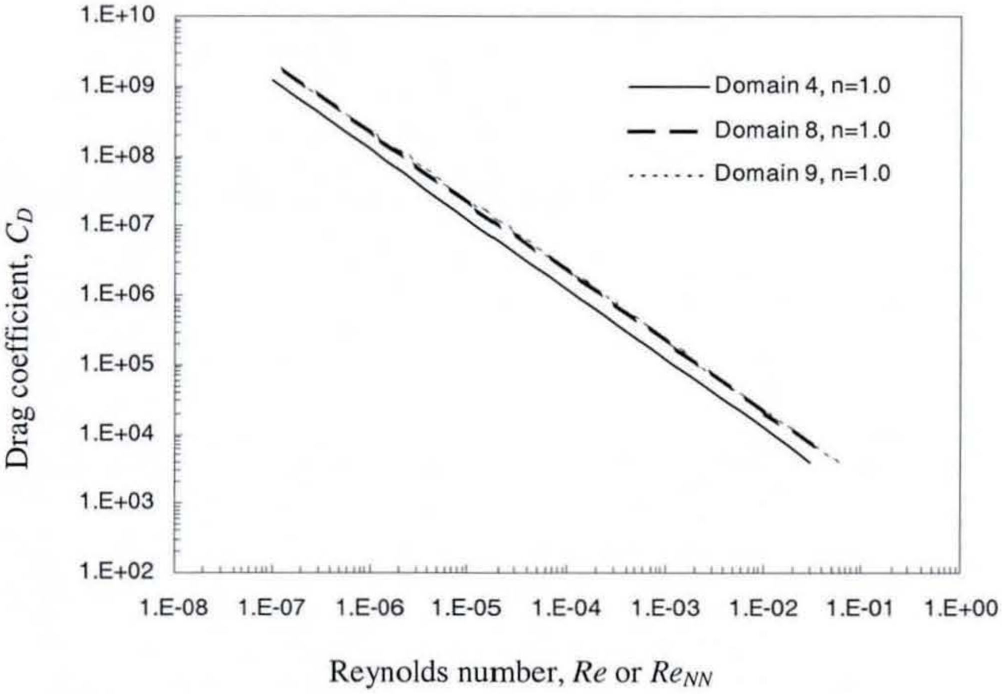


Figure 5.66 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domains 4, 8 and 9.

5.13 Pressure Drop Analysis

To further verify the reliability of the developed model, the results obtained for fluid flow through domain 1 were tested against experimental data extracted from experimental studies conducted by Rushton (1969) using water and Chhabra and Richardson (1985) using shear thinning fluids. The physical properties of the test fluids and the characteristic dimensions of the wire mesh and monofilament cloths used in Chhabra and Richardson's (1985) and Rushton's (1969) experiments and in this study are summarized in Table 5.5.

Table 5.5 Physical properties of the test fluids and characteristic dimensions of the plain weave wire mesh screens and monofilament cloth.

Test Liquid	n	η_0 (Pa s)	Aperture (μm)	d (μm)	Symbols in Figure 5.67
Chhabra and Richardson (1985); wire mesh screens					
1.5% CMC in water	0.60	6.6	53	36	■
1.5% CMC in water	0.59	7.6	53	36	◆
2.0% CMC in water	0.61	9.5	150	100	▲
2.5% CMC in water	0.34	45.0	150	100	●
Rushton (1969); monofilament Nylon cloth					
Water	1.0	0.001	24	30	+
	1.0	0.001	42.4	37	+
	1.0	0.001	60	30	+
	1.0	0.001	59	44	+
	1.0	0.001	71	61	+
	1.0	0.001	99	75	+
	1.0	0.001	144	105	+
	1.0	0.001	186	153	+
This study					
1.5% CMC in water	0.60	6.6	300	200	□
1.5% CMC in water	0.59	7.6	300	200	◇
2.0% CMC in water	0.61	9.5	300	200	△
2.5% CMC in water	0.34	45.0	300	200	○
Water	1.0	0.001	300	200	×

It has been explained earlier that Chhabra and Richardson (1985) used an erroneous parameter in their equation for Reynolds number. Therefore, the experimental results presented in their paper were recalculated using equations (2.6) and (5.1) and presented in Figure 5.67. The results for water flow through monofilament nylon cloths reported by Rushton (1969) were also calculated using equations (2.6) and (5.1) and included on Figure 5.67. From Figure 5.67, the simulation results for water and the non-Newtonian liquid were observed to be in good agreement with the experimental results. Furthermore, the drag coefficient C_D and the Reynolds number Re_{NN} are related by a best fit curve of

$$C_D = \frac{10}{\mathrm{Re}_{NN}{}^{1.14}} \tag{5.2}$$

The C_D versus Re_{NN} plot for Newtonian, shear thickening and shear thinning fluid flow through domains 1 to 4 were plotted in Figures 5.68, 5.69 and 5.70, respectively. Similar plots for domains 5 to 8 were plotted in Figures 5.71 to 5.73.

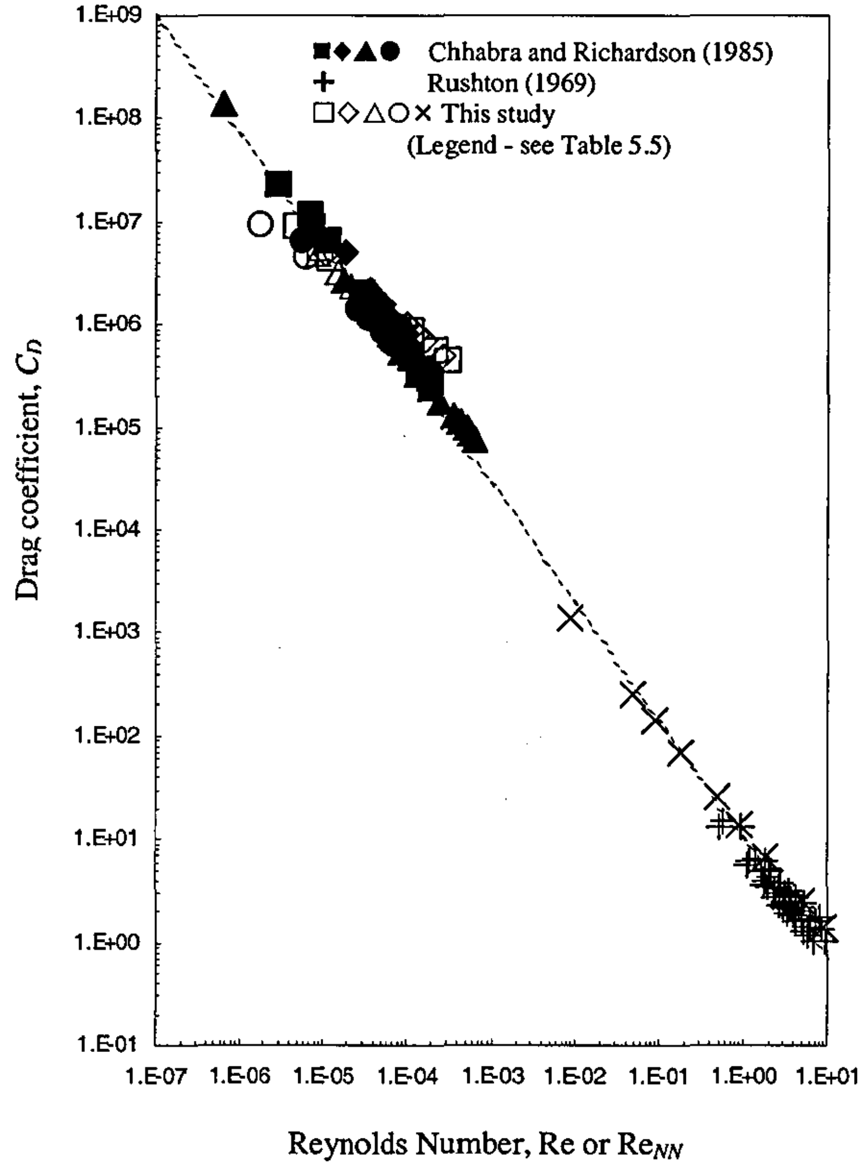


Figure 5.67 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian liquid flows through domain 1.

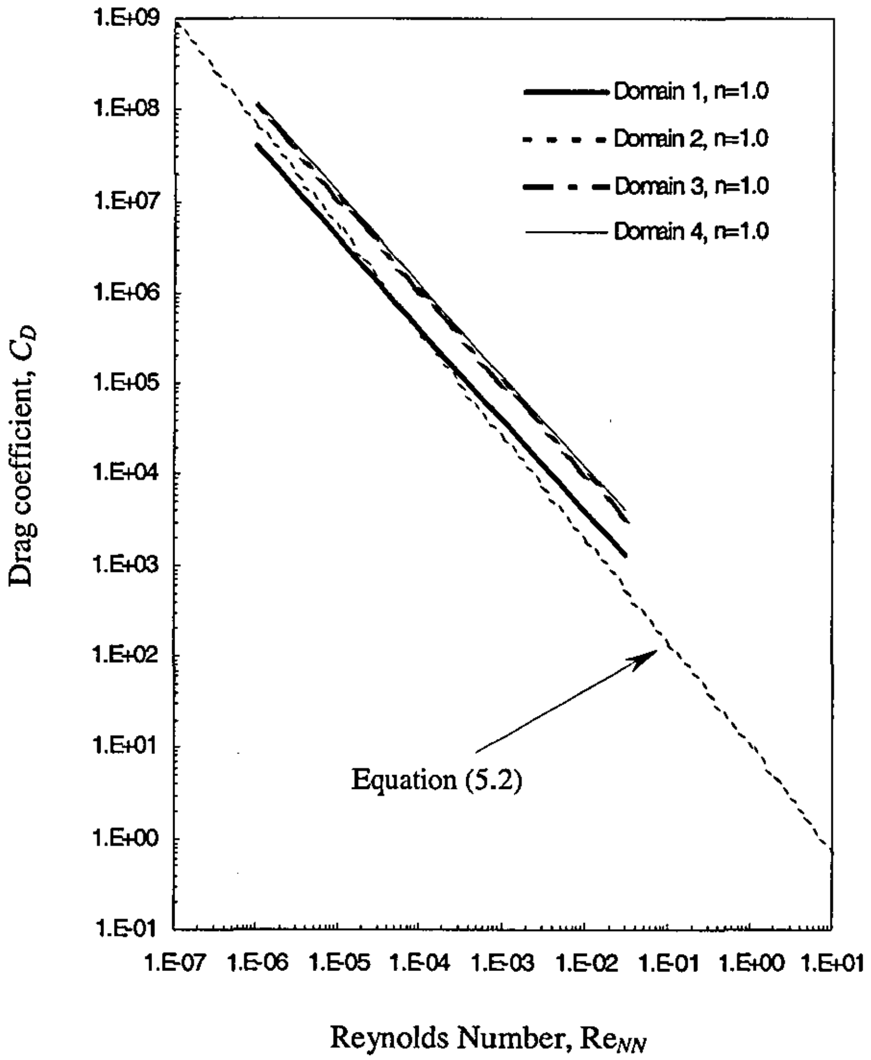


Figure 5.68 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domains 1, 2, 3 and 4.

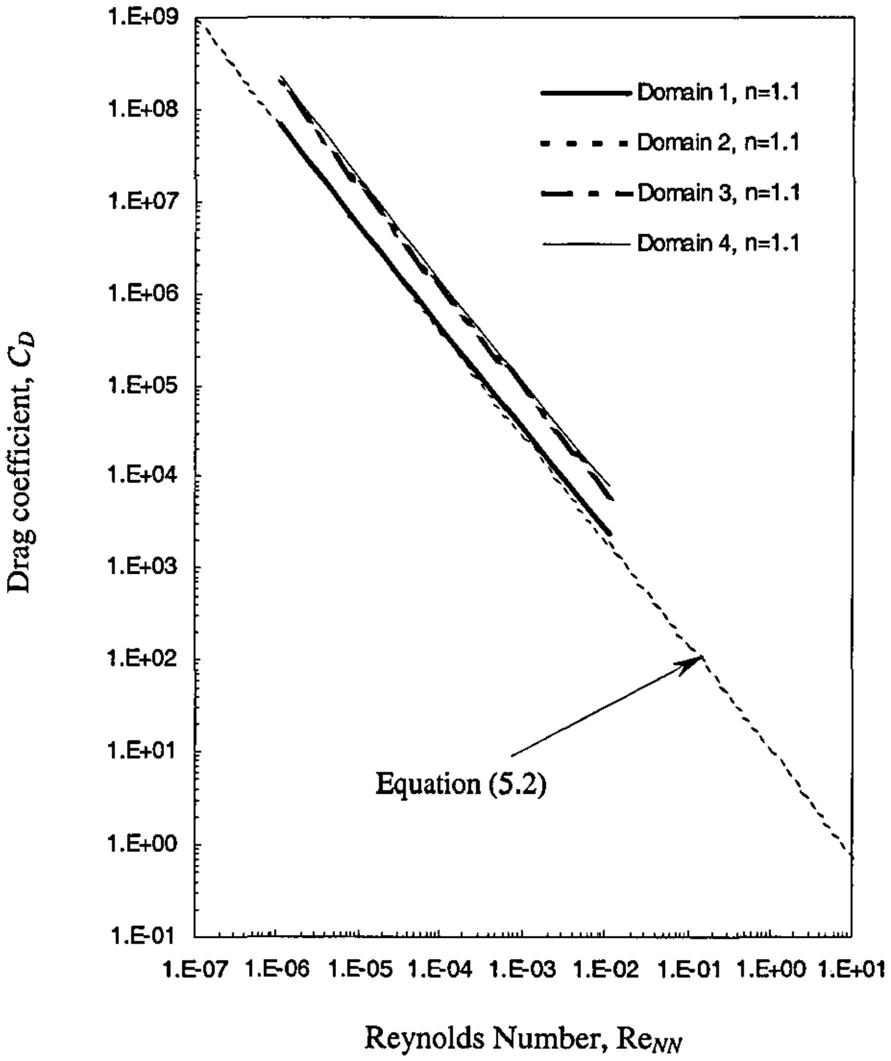


Figure 5.69 Drag Coefficient *vs* Reynolds Number for shear thickening fluid flow through domains 1, 2, 3 and 4.

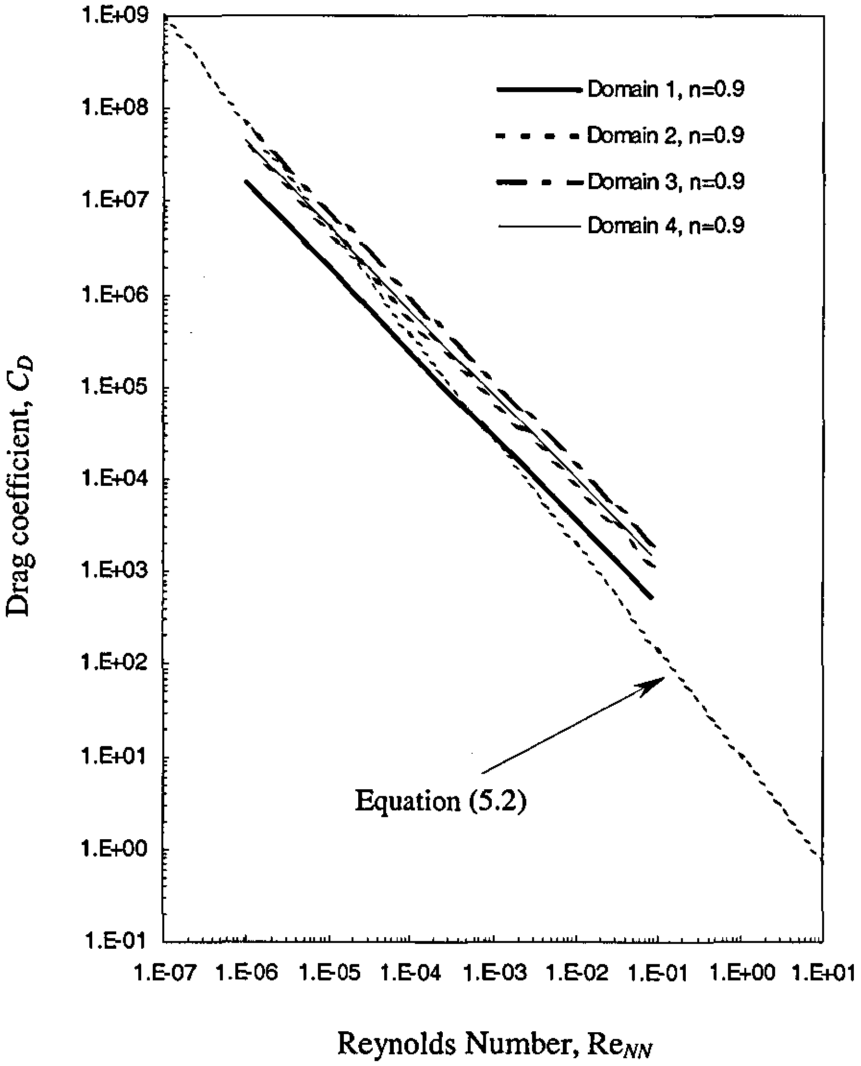


Figure 5.70 Drag Coefficient *vs* Reynolds Number for shear thinning fluid flow through domains 1, 2, 3 and 4.

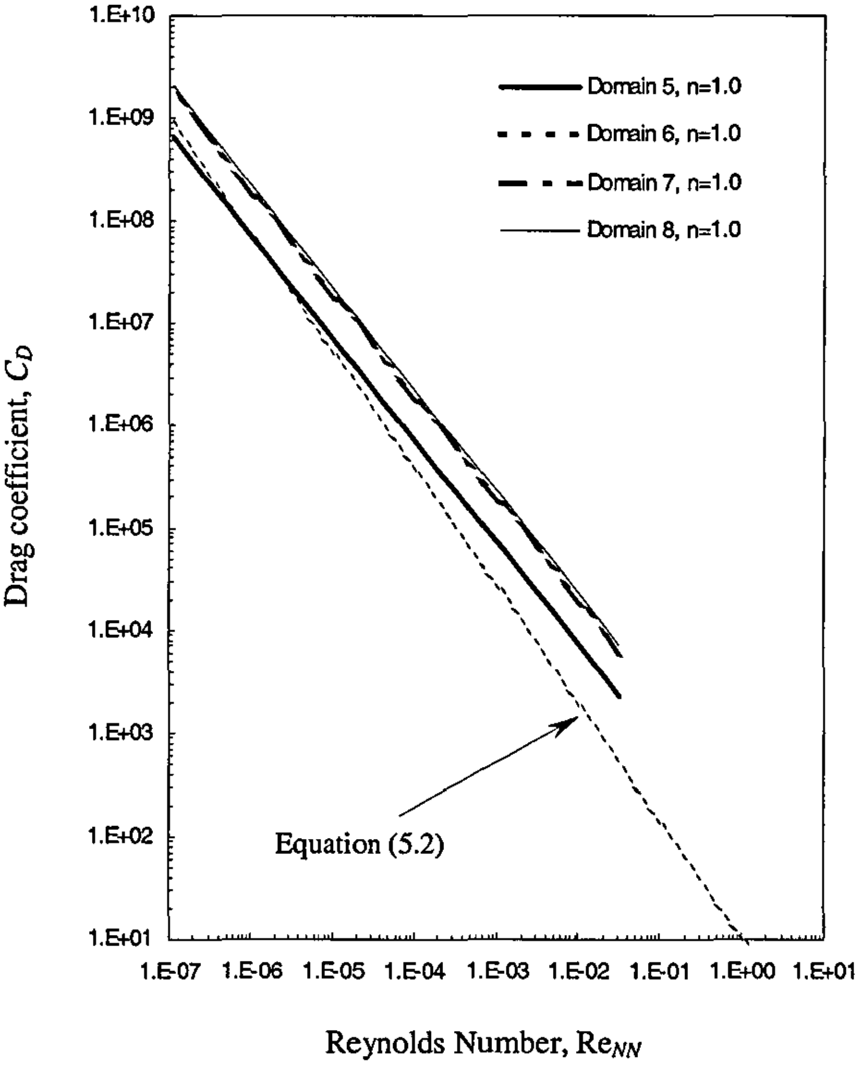


Figure 5.71 Drag Coefficient *vs* Reynolds Number for Newtonian fluid flow through domains 5, 6, 7 and 8.

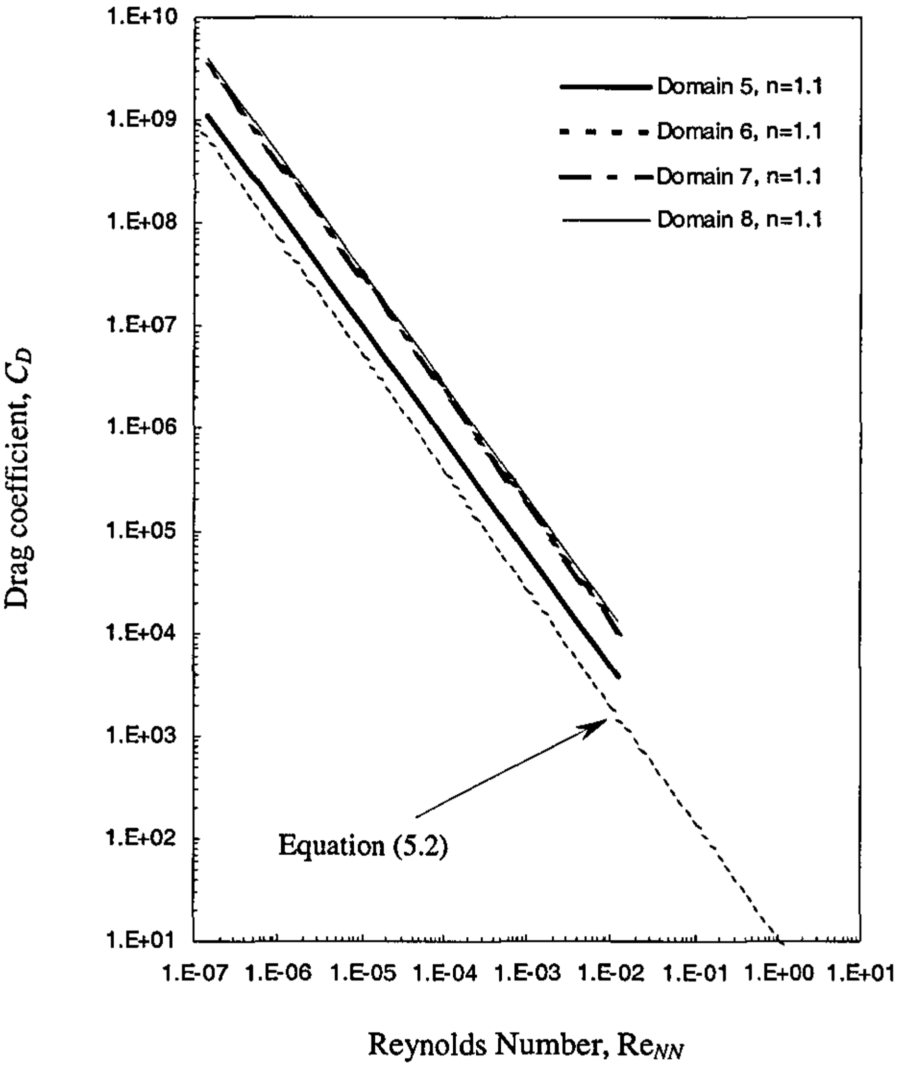


Figure 5.72 Drag Coefficient *vs* Reynolds Number for shear thickening fluid flow through domains 5, 6, 7 and 8.

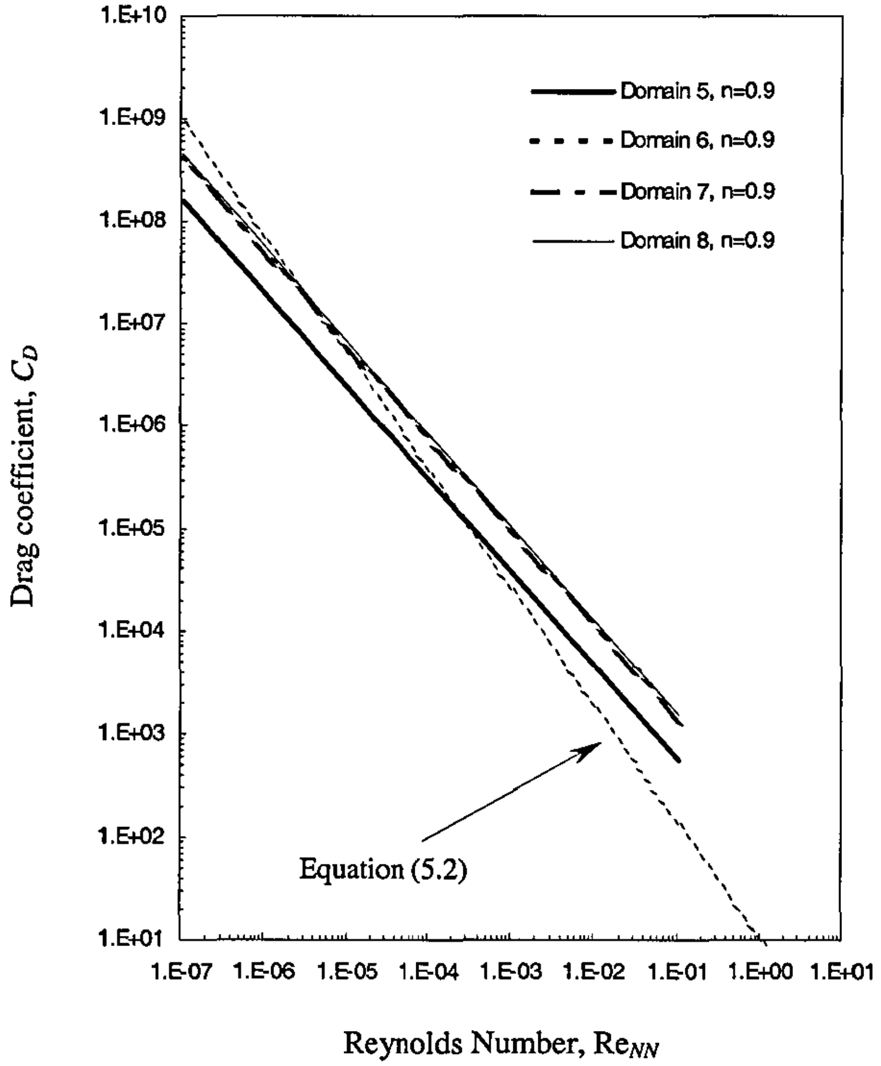
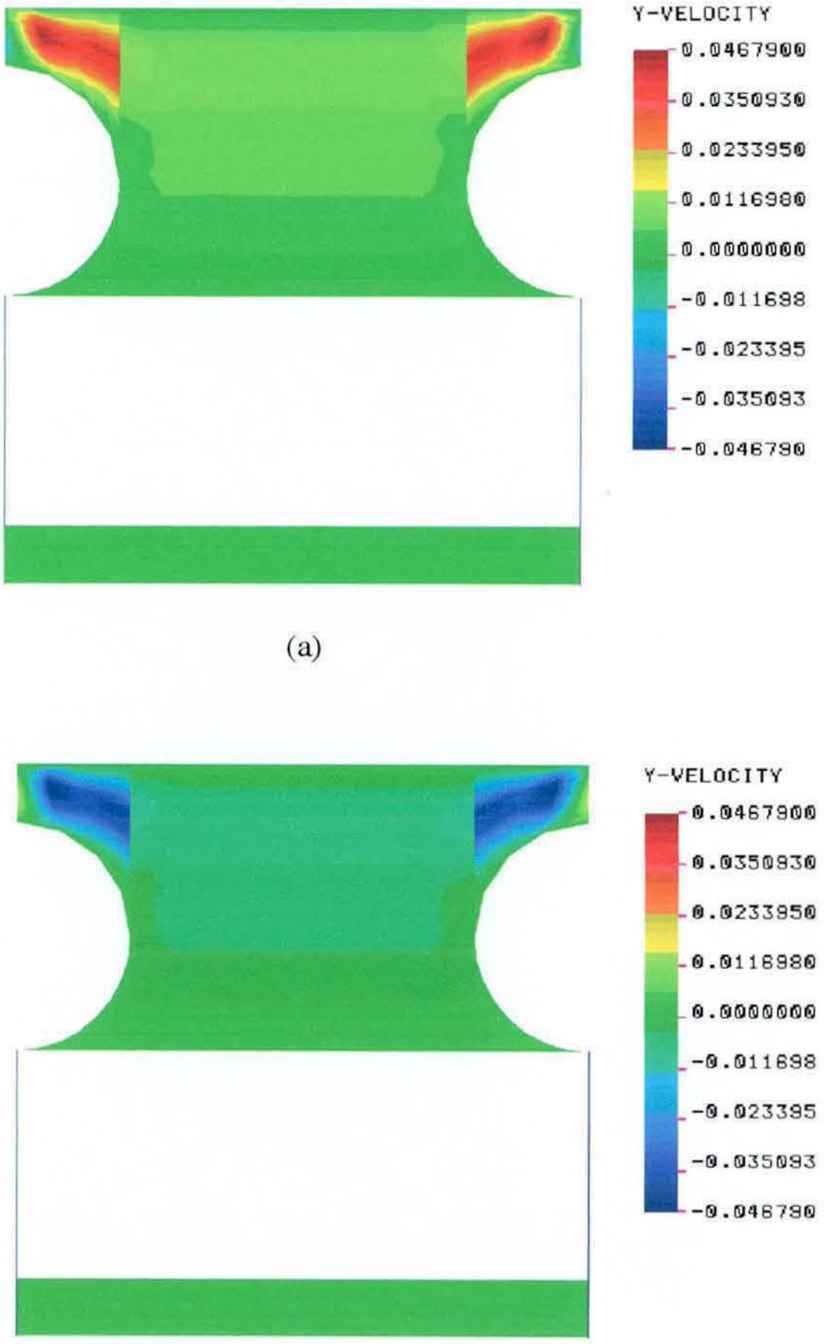


Figure 5.73 Drag Coefficient *vs* Reynolds Number for shear thinning fluid flow through domains 5, 6, 7 and 8.

From Figures 5.68 to 5.73, it can be observed that pore type 1 (represented by domains 1 and 5) consistently gives the lowest pressure drop, while pore type 4 (represented by domains 4 and 8) gives the highest pressure drop across the wire mesh. The only exception is Figure 5.70 where pore type 3 was marginally higher in pressure drop compared to pore type 4. This observation however is in serious disagreement to the observation by Lu *et al.* (1996) and Tung *et al.* (2002) as they reported the lowest pressure drop for pore type 4 and highest pressure drop for pore type 1. To explain this discrepancy, the experimental data of previous researchers and industrial wire mesh manufacturers were consulted. A closer examination of the experimental results reported by Armour and Cannon (1968) and Rushton and Griffiths (1971) showed a consistently lower pressure drop for plain weave compared to twilled weave, with satin weave ranging between them. As plain weave has the most uniform pore type amongst other weave patterns, where it consists entirely of pore type 1, plain weave can be safely used as a benchmark to check the accuracy of the numerical model. The consistency of our results with those of Armour and Cannon (1968) and Rushton and Griffiths (1971) showed the accuracy of our predictions. Personal communication with industrial experts and wire mesh manufacturers have confirmed the industrial practice whereby plain weave is primarily used due to its lowest flow resistance. These cross-examinations confirmed the validity of our results and gives confidence to our model.

Finally, it should be mentioned that the geometry of simulation domains allow the fluid to cross through openings in the artificial vertical boundaries as it flows through the interstices. This is a deliberate decision taken to preserve the actual physics of flow around the monofilament wire fiber and in the interstices between the wires. This however implied the examination of mass continuity has been taken to a greater complexity. Figure 5.74 illustrates the y-velocity component taken in x-z plane at $y = l_w$ and $y = 0$. The symmetrical plots showed the expected outflow of fluid normal to the x-z plane at both artificial boundaries with the magnitude of the fluid flow being identical, albeit in opposite y- direction. Since the validation of the continuity has been addressed in the test cases, the representation of the fluid flow structure in the interstices has been considered more significant in a three-dimensional study such as the present investigation.



(a)

(b)

Figure 5.74 The y- velocity component of Newtonian fluid flow through domain 4 in the x-z plane at (a) $y = l_w$ and (b) $y = 0$ with inlet velocity v_z = -0.01ms^{-1}.

5.14 Closure

In this chapter, a selection of computational domains has been used to investigate the fluid flow through four basic pore types in a plain weave, twill weave and satin weave. The effect of weave pattern, aperture to diameter ratio and Reynolds number on flow pattern and pressure drop has been systematically studied. The results for each simulation domains were presented and discussed. The analysis has shown strong evidence that the developed model is capable of generating accurate results for flow of Newtonian and non-Newtonian fluids flow through monofilament filter media.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

6.1 Conclusions

Three-dimensional weighted residual finite element schemes for solving Stokes flow have been successfully developed and applied to simulate flow through monofilament wire meshes. Two robust and reliable computer tools based on the sound and robust numerical technique mentioned have been developed to simulate Newtonian and non-Newtonian fluid flow through a woven wire mesh. The governing equations of continuity and momentum were solved by a mixed U-V-W-P finite element method in conjunction with a first order Taylor-Galerkin scheme for temporal discretization. A secondary solution scheme based on a continuous Penalty finite element method in conjunction with theta time stepping method was also used to solve the governing equations. The accuracy of the model is proven by three purpose-designed test cases of increasing complexity and compared against experimental data from the literature for simulation of flow through wire meshes.

A selection of domains was used to investigate the effect of weave pattern, aperture to diameter ratio and Reynolds number on flow pattern and pressure drop. Based on these domains, simulations were successfully conducted to investigate fluid flow through four basic pore types in a plain weave, twill weave and satin weave. The flow fields in the interstices were illustrated using commercial graphics software package. The results successfully showed the weave pattern has a profound effect on the fluid flow pattern and pressure drop across the wire mesh.

The pressure drop across the wire mesh was analyzed and the results were found to be consistent with the existing experimental literature. Simulation results have shown that pore type 1 gives the lowest pressure drop, while pore type 4 gives the highest pressure drop across the woven cloths. This is consistent with the experimental results reported by Armour and Cannon (1968) and Rushton and Griffiths (1971), showing the accuracy of the predictions in this study. Personal communication with industrial experts and woven cloths manufacturer has confirmed the industrial practice whereby plain weave, which consists primarily of pore type 1, is primarily used due to its lowest flow resistance.

To further verify the reliability of the developed model, the results obtained for fluid flow through domain 1 were tested against experimental studies by Rushton (1969) using water and Chhabra and Richardson (1985) using shear-thinning fluids. The drag coefficient versus Reynolds number plots based on the experimental results and the predictions in this study were found to be in close agreement.

This showed that the developed model is capable of generating accurate results for flow of both Newtonian and non-Newtonian fluids through filter media. The model can be used by design engineers as a convenient and effective Computer Aided Design (CAD) tool for quantifying effects of pressure drop. The model can be extended to describe particle capture on/in the domains of wire mesh and woven filter cloths. As the model was developed with a high degree of flexibility, it is suitable for general applications in solving a wide range of flow systems in three-dimensional domains. These codes are currently being used by other PhD researchers in the group for various investigations including bio-medical applications such as the development of artificial heart valve.

6.2 Recommendations for Future Work

This study has provided strong evidence that the developed algorithms can be used to predict fluid flow through a geometrically complex domain. The results provide a sound basis to proceed with future investigations. The continuation of this work can be extended to examine a number of effects not covered in this study.

(1) To extend the individual unit adopted in this study to examine the effect of fluid flow through a series of combined individual units. The interaction of fluid flow across neighbouring units as it flows through the interstices can shed further light on the flow distributions and pressure drop in the interstices of a series of pore units.

(2) To examine the effect of different pore types combined in a single solution domain. This is significant as in weaves such as the twill weave and satin weave, the woven cloth normally has two to three basic pore types in a unit area. The proportion of the basic pore types can be carefully chosen to emulate a typical twill or satin weave.

(3) The study can be extended to more complicated weaves such as the plain Dutch weave, Dutch twilled weave and reversed plain Dutch weave. The absence of an aperture in the direction of fluid flow will undoubtedly pose a challenge in numerical simulations and pressure drop analysis. The flow distribution is expected to be complex and the pressure drop analysis is also expected to be not

straightforward. However, since this category of weaves has been widely used in industrial applications, it is only reasonable to see this area further explored.

(4) The study can also be extended to study the effect of overlapping layers of screens. Sintered composite wire meshes have been widely used in the industry and the study to effectively quantify the flow and the pressure drop across the layers will be an interesting prospect. For example, studies can be conducted on combined layers of similar aperture and diameter ratio, and combined layers of differing aperture size and diameter ratio. In any case, the effect of distance between these layers on pressure drop and flow distributions should be investigated concurrently.

(5) The model can be extended to incorporate particle capture in/on the wire mesh. This will be an interesting subject for processes involving either surface filtration or depth filtration. In particularly, the initial stages of filter cake formation in surface filtrations can be studied to improve the performance of the filter media. Furthermore, the efficiency of various types and specifications of wire meshes for filtering particulates, colloids and macromolecules can be examined and evaluated. This can be achieved by introducing an additional subroutine solving convective-diffusion type equations.

(6) Metallic wire meshes have been widely used in heat transfer applications; for example, as a regenerator in Stirling cryocoolers, in waste heat recovery units of gas turbines and in the enhancement of cooling of electrical and electronic equipments. The heat transfer mechanism can be incorporated into the developed model and studied systematically.

(7) A software package with user-friendly graphic interfaces can be generated to facilitate industrial design engineers and process engineers. With easy punch-in of parameters such as aperture, wire diameter and flow rate, an estimated pressure drop over a wire mesh systems (and *vice versa*) can be given based on the correlations obtained from the study. With available IT tools containing front and back end interfaces and support, it will undoubtedly be a significant contribution to the industry with wire mesh applications.

REFERENCES

Argyris, J. H., 1960. Energy theorems and structural analysis, Butterworth (reprinted from *Aircraft Eng.*, 1954-1955).

Armour, J. C. and Cannon, J. N., 1968. Fluid flow through woven screens. *AIChE Journal*, **14**(3), 415-420.

Astarita, G., 1976. Is non-Newtonian fluid mechanics a culturally autonomous Subject? *J. non-Newtonian Fluid Mech.*, **1**, 203-206.

Astarita, G. and Marrucci, G., 1974. Principles of non-Newtonian fluid mechanics, New York: Mc-Graw Hill.

Babuška, I., 1971. Error-bounds for finite element method. *Numer. Math.*, **16**, 322-333.

Babuška, I., Banerjee, U. and Osborn, J. E., 2002. On principles for the selection of shape functions for the generalized finite element method. *Comput Methods Appl. Mech Engrg.*, **191**, 5595-5629.

Babuška, I. and Narasimhan, R., 1997. The Babuska-Brezzi condition and the patch test: an example. *Comput Methods Appl. Mech Engrg.*, **140**, 183-199.

Babuška, I., and Strouboulis, T., 2001. The finite element method and its reliability. Oxford University Press, New York.

Backer, S., 1948. The relationship between the structural geometry of a textile fabric and its physical properties. Part I: literature review. *Text. Res. J.*, **18**, 650-658.

Backer, S., 1951. The relationship between the structural geometry of a textile fabric and its physical properties, Part IV: Interstice geometry and air permeability. *Text. Res. J.*, **21**, 703-714.

Baker, A., 1956. The mechanical properties and porosity of parachute fabrics. RAE Report No. Chem 507 (AD 123 193).

Bathe, K. J., 1996. Finite element procedures, Englewood Cliffs, NJ: Prentice Hall.

Bathe, K. J., 2001. The inf-sup condition and its evaluation for mixed finite element methods. *Comput. Struct.*, **79**, 243-252.

Bathe, K. J. and Iosilevich, A, Chapelle, D., 2000. An inf-sup test for shell finite elements. *Comput. Struc.*, **75**, 439-456.

Bercovier, M. and Engelman, M., 1979. A finite element for the numerical solution of viscous incompressible flows. *J. Comput. Phys.*, **30** (2), 181-201.

Bey, K. S., Thornton, E. A., Dechaumphai, P. and Ramakrishnan, R., 1985. A new finite element approach for prediction of aerothermal loads:progress in inviscid flow computations. *AIAA 7th Computational Fluids Dynamics Conference*, Cincinnati, OH, July 15-17, 411-424.

Bird, R. B., Stewart, W. E. and Lightfoot, E. N., 2002. Transport phenomena, 2nd ed., Wiley, USA.

Brezzi, F., 1974. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *RAIRO Analyse Numerique* 8R-2, 129-151.

Brezzi, F. and Pitkäranta, J., 1984. On the stabilization of finite element approximations of the Stokes problem. In W. Hackbusch, editor, *Efficient solution of Elliptic Problems, Notes on Numerical Fluid Mechanics*, **10**, Vieweg, Wiesbaden.

Chapelle, D., Bathe, K. J., 1993. The inf-sup test. *Comput Struc,* **47**, 537-545.

Chhabra, R. P., 1993a. Bubbles, drops, and particles in non-Newtonian fluids. *Boca Raton, FL:CRC Press.*

Chhabra, R. P., 1993b. Fluid flow, heat and mass transfer in non-Newtonian fluids: Multiphase systems. *Advances in Heat Transfer*, **23**, 187-278.

Chhabra, R. P., 1993c. Transport process in particulate systems with non-Newtonian fluids. *Advances in Transport Processes*, **9**, 501-577.

Chhabra, R. P., 1999. Laminar boundary layer heat transfer to power-law fluid: An approximate analytical solution. *Journal of Chemical Engineering of Japan*, **32**, 812-816.

Chhabra, R. P., Comiti, J. and Machač, I., 2001. Flow of non-Newtonian fluids in fixed and fluidized beds : review. *Chem. Eng. Sci.*, **56**, 1-27.

Chhabra, R. P. and Richardson, J. F., 1985. Flow of liquids through screens: Relationship between pressure drop and flow rate. *Chem. Eng Sci*, **40**(2), 313-316.

Chhabra, R.P. and Richardson, J. F., 1999. Non-Newtonian flow in the process industries, Oxford, UK: Butterworth-Heinemann.

Clough, R. W., 1960. The finite element in plane stress analysis. *Proc. 2nd ASCE Conf. On Electronic Computation*, Pittsburgh, Pa., Sept.

Courant, R., 1943. Variational methods for the solution of problems of equilibrium and vibration. *Bull. Amer. Math. Soc.*, **49**, 1-61.

Crochet, M. J., Davies, A. R. and Walters, K., 1984. Numerical simulation of non-Newtonian flow, rheology series 1, elsevier, Amsterdam.

Davies, C. N., 1973. Air filtration. Academic Press, New York.

Demkowicz, L., Devloo, P. and Oden, J. T., 1985. On an h-type mesh refinement strategy based on minimization of interpolation errors. *Comput. Methods Appli. Mech. Eng.*, **53**, 67-89.

Demkowicz, L., Oden, J. T., Rachowicz, W. and Hardy, O., 1989. Toward a universal *h-p* adaptive finite element strategy, 1: constrained approximation and data structures. *Comput. Methods Appl. Mech. Eng.*, **77**, 79-112.

Das, D. B., Nassehi, V. and Wakeman, R. J., 2002. A finite volume model for the hydrodynamics of combined free and porous flow in sub-surface regions. *Advances in Environmental Research,* **7**, 35-58.

Dolej, V., and Siska, B., 2000. Flow of viscoelastic fluid through fixed beds of particles: comparison of three approaches. *Chemical Engineering and Processing,* **39**, 417-423.

Donea, J., 1984. A Taylor-Galerkin method for convective transport problems. *Int. J. Numer. Methods Eng.*, **20**, 101-119.

Drummond, J. E. and Tahir, M. I., 1984. Laminar viscous flow through regular arrays of parallel solid cylinders. *International Journal of Multiphase Flow,* **10**, 515-534.

Ehrhardt, G., 1983. Flow measurements for wire gauzes. *International Chemical Engineering*, **23**, 455-465.

Faxén, O. H., 1946. Forces exerted on a rigid cylinder in a viscous fluid between two parallel fixed planes. *Proc. R. Swed. Acad. Eng. Sci.* **187**, 1-13.

Ghoreishy, M. H. R. and Nassehi, V., 1997. Modelling the transient flow of rubber compounds in the dispersive section of an internal mixer with slip-stick boundary conditions. *Adv. Poly. Tech.* **16**, 45-68.

Goglia, M. J., LaVier, H. W. S. and Brown, C. D., 1955. Air permeability of parachute cloths. *Textile Research Journal*, **25**(4).

Gresho, P. M., Lee, R. L. and Sani, R. L., 1980. On the time-dependent solution of the incompressible Navier-Stokes equations in two and three dimensions. In *Recent Advances in Numerical Methods in Fluids,* Pinebridge Press Limited, Swansea.

Happel, J. and Brenner, H., 1973. Low Reynolds number hydrodynamics, 2nd edition, Noordhoff, Leyden, Netherlands.

Harris, John, 1977. Rheology. In: Rheology and non-Newtonian flow, ch. 1, Longman, Great Britain.

Hartt, W. H., and Baird, D. G., 1996. The confined flow of polyethylene melts past a cylinder in a planar channel. *J. Non-Newtonian Fluid Mech.*, **65**, 247-268.

Heinrich, J. C. and Pepper, D. W., 1999. Intermediate Finite Element Method: Fluid flow and heat transfer applications. *Series in computational and physical processes in mechanics and thermal science*, Taylor & Francis, New York.

Huang, H. C., Li, Z. H. and Usmani, A. S., 1999. Finite element analysis of non-Newtonian flow. Springer, London.

Huebner, Kenneth H., Dewhirst, Donald L., Smith, Douglas E. and Byrom, Ted G., 2001. The Finite Element Method for Engineers, 4th edition, John Wiley & Sons, Inc., USA.

Hughes, T. J. R., Liu, W. K. and Brooks, A., 1979. Finite element analysis of incompressible viscous flows by the Penalty function formulation. *J. Comput. Phys.*, **30**(1), 1-60.

Ingmanson, W. L., Han, S. T., Wilder, H. D. and Myers, W. T., 1961. Resistance of Wire Screens to Flow Water. *TAPPI*, **44**, 47-54.

Iosilevich, A., Bathe, K. J. and Brezzi, F., 1996. Numerical inf-sup analysis of MITC plate bending elements. In *Proc. Of the American Math. Society, Symposium on Plates and Shells.* Universite Laval.

Iosilevich, A, Bathe, K. J and Brezzi, F., 1997. On evaluating the inf-sup condition for plate bending elements. *Int J Num Meth Eng*, **40**, 3639-3663.

Kemblowski, Z., Dziubinski, M. and Sek, J., 1989. Flow of non-Newtonian fluids through granular media. In Mashelkar, R.A., A.S. Mujamdar, and R. Kamal (eds.), *Transport phenomena in polymeric systems.* Chichester: Ellis Horwood.

Kiljański, T. and Dziubiński, M., 1996. Resistance to flow of molten polymers through filtration screens. *Chem. Eng. Sci.*, **51**(19), 4533-4536.

Ladyzhenskaya, O. A., 1969. The mathematical theory of viscous incompressible flow. New York: Gordon and Breach.

Lapidus, L. and Pinder, G. F., 1982. *Numerical solution of partial differential equations in science and engineering*, Wiley, New York.

Liu, A. W., Bornide, D. E., Amstrong, R. C. and Brown, R. A., 1998. Viscoelastic flow of polymeric solutions around a periodic, linear array of cylinders:comparisons of predictions for microstructure and flow fields. *J. Non-Newtonian Fluid Mech.*, **77**, 153-190.

Löhner, R., Morgan, K. and Zienkiewicz, O. C., 1984a. The solution of non-linear systems of hyperbolic equations by the finite element method. *Int. J. Numer. Method Fluid*, **4**, 1043-1063.

Löhner, R., Morgan, K. and Zienkiewicz, O. C., 1984b. The use of domain splitting with explicit hyperbolic solver, *Comput. Methods Appl. Mech. Eng.*, **45**, 313-329.

Löhner, R., Morgan, K. and Zienkiewicz, O. C., 1985. An adaptive finite element procedure for compressible high speed flows. *Comput. Methods Appl. Mech. Eng.*, **51**, 441-465.

Löhner, R., 1987. An adaptive finite element scheme for transient problems in CFD. *Comput. Methods Appl. Mech. Eng.*, **61**, 323-338.

Lu, Wei-Ming, Tung, Kuo-Lun and Hwang, Kuo-Jen, 1996. Fluid flow through basic weaves of monofilament filter cloth. *Textile Res. J.*, **66**(5), 311-323.

Mauret, E. and Renaud, M., 1997. Transport phenomena in multi-particle systems : I. Limits of applicability of capillary model in high voidage beds-application to fixed beds of fibers and fluidized beds of spheres. *Chem. Eng. Sci.*, **52**, 1807-1817.

Mauret, E. and Renaud, M., 1997a. Transport phenomena in multi-particle systems-I. Limits of applicability of capillary model in high voidage beds – application to fixed beds of fibers and fluidized beds of spheres. *Chem. Eng. Sci.*, **52**, 1807-1817.

Mauret, E. and Renaud, M., 1997b. Transport phenomena in multi-particle systems-II. Limits of applicability of capillary model in high voidage beds – application to fixed beds of fibers and fluidized beds of spheres. *Chem. Eng. Sci.*, **52**, 1807-1817.

McCabe, W. L., Smith, J. C. and Harriott, P., 2001. Unit operations of chemical engineering, 6th edition, McGrawHill, NY.

Metzner, A. B., 1956. Non-Newtonian Technology : Fluid Mechanics, mixing, and heat transfer, in *Advances in Chemical Engineering*, 1, edited by Drew T.B. and Hooper J.W. Academic Press, New York.

Metzner, A. B., 1961. Flow of non-Newtonian fluids, in *Handbook of Fluid Dynamics*, first edition, edited by Streeter V.L, McGraw-Hill Inc., New York.

Mitsoulis, E., 1998. Numerical simulation of confined flow of polyethylene melts around acylinder in a planar channel. *J. Non-Newtonian Fluid Mech.*, **76**, 327-350.

Nakazawa, S., Pittman, J. F. T. and Zienkiewicz, O. C., 1982. Numerical solution of flow and heat transfer in polymer melts. In Gallagher, R.H., et al. (eds), *Finite Elements in Fluids*, 4, Ch 13, Wiley, Chichester.

Nassehi, V., 1998. Modelling of combined Navier-Stokes and Darcy flows in crossflow membrane filtration. *Chem. Eng. Sci.*, **53**(6), 1253-1265.

Nassehi, V., 2002. Practical aspects of finite element modelling of polymer processing. Wiley, Chichester, UK.

Nassehi, V., Hanspal, N. S., Waghode, A. N., Ruziwa, W. R. and Wakeman, R. J., 2005. Finite-element modelling of combined free/porous flow regimes: simulation of flow through pleated cartridge filters. *Chem. Eng. Sci.*, **60**(4), 995-1006.

Oden, J. T., Strouboulis, T. and Devloo, P., 1986. Adaptive finite element methods for the analysis of inviscid compressible flow, I: fast refinement/unrefinement and moving mesh methods for unstructured meshes. *Comput. Methods Appl. Mech. Eng.*, **59**, 327-362.

Oden, J. T., Demkowicz, L., Rachowicz, W. and Westermann, T. A., 1989. Toward a universal *h-p* adaptive finite element strategy, 2: A posteriori error estimation. *Comput. Methods Appl. Mech. Eng.*, **77**, 113-180.

Oñate, E., 1998. Derivation of stabilized equations for numerical solution of advective-diffusive transport and fluid flow problems. *Comput. Meth. Appli. Mech. Engng*, **151**, 233-265.

Peano, A. G., 1976. Hierarchies of conforming finite elements for plan elasticity and plate bending. *Comput. Math. Appl.*, **2**, 211-224.

Peano, A. G., Passini, A., Riccioni, R. and Sardella, L., 1979. Adaptive approximation in finite element structural analysis. *Comput. Struct.*, **10**, 332-342.

Pederson, G. C., 1969. Fluid flow through monofilament fabrics. Paper presented at 64th National Meeting of A.I.Ch.E., New Orleans.

Pittman, J. F. T. and Nakazawa, S., 1984. Finite element analysis of polymer processing operations. In: Pittman, J.F.T., Zienkiewicz, O.C., Wood, R.D. and Alexander, J.M. (eds), *Numerical Analysis of Forming Processes*, Ch 6, Wiley, Chichester.

Rachowicz, W., Oden, J. T. and Demkowicz, L., 1989. Toward a universal *h-p* adaptive finite element strategy, 3: design of *h-p* meshes. *Comput. Methods Appl. Mech. Eng.*, **77**, 181-212.

Rahli, O., Tadrist, L. and Miscevic, M., 1996. Experimental analysis of fibrous media permeability. *A.I.Ch. E. Journal*, **42**, 3547-3549.

Rainard, L. W., 1946. Air permeability of fabrics. *Textile Research Report*, **16**(10).

Rainard, L. W., 1947. Air permeability of fabrics II. *Textile Research Report*, **17**(3).

Ramakrishnan, R., Bey, K. S. and Thornton, E. A., 1990. Adaptive quadrilateral and triangular finite element scheme for compressible flows. *AIAA J.*, **28**(1), 51-59.

Ramakrishnan, R., Wieting, A. R. and Thornton, E. A., 1992. A transient finite element adaptation scheme for thermal problems with steep gradients. *Int. J. Numer. Methods Head Fluid Flow*, **2**, 517-535.

Rao, T. V. M. and Chhabra, R. P., 2003. A note of pressure drop for the cross-flow of power-law liquids and air/power law liquid mixtures past a bundle of circular rods. *Chem. Eng. Sci.*, **58**, 1365-1372.

Reddy, J. N., 1982. On Penalty function methods in the finite element analysis of flow problems. *Int. J. Numer. Methods Fluid*, **2**, 151-171.

Reddy, J. N., 1982. Penalty-Finite-Element analaysis of 3-D Navier-Stokes Equations. *Comput. Methods Appl. Mech. Eng.*, **35**, 87-106.

Richardson, C. J. and Nassehi, V., 2003. Finite Element Modelling of Concentration Profiles in Flow Domains With Curved Porous Boundaries. *Chem. Eng. Sci.* **58**(12), 2491-2503.

Robertson, A. F., 1950. Air porosity of open weave fabrics. *Text. Res. J.*, **20**, 838.

Rushton, A., 1969. Filtration Research Report. Chem. Eng. Dept., UMIST.

Rushton, A. and Griffiths, P., 1971. Fluid Flow in Monofilmant Filter Media. *Trans Instn Chem. Engrs*, **49**, 49-59.

Satheesh, V. K., Chhabra, R. P. and Eswaran, V., 1999. Steady incompressible fluid flow over a bundle of cylinders at moderate Reynolds numbers. *Canadian Journal of Chemical Engineering*, **77**, 978-987.

Schowalter, W. R., 1978. Mechanics of non-Newtonian fluids, Oxford, UK: Pergamon.

Shibu, S., Chhabra, R. P. and Eswaran, V., 2001. Power law fluid flow over a bundle of cylinders at intermediate Reynolds numbers. *Chem. Eng. Sci.*, **56**, 5545-5554.

Silliman, W. J. and Scriven, L. E., 1980. Separating flow near a static contact line:slip at a wall and shape of a free surface. *J. Comput. Phys.* **34**, 287-313.

Skartsis, L., Khormami, B. and Kardos, J. L., 1992. Polymeric flow through fibrous media. *Journal of Rheology*, **36**, 589-608.

Skelland, A. H. P., 1967. Non-Newtonian flow and heat transfer, Wiley, New York, USA.

Skelton, J. and Abbott, N. J., 1974. Development of stretch fabric for parachute canopies. Technical report ASD-TR-74-26, Wright-Patterson Air Force Base, Ohio (AD 785 195).

Squiers, J. C., 1984. Fluid flow resistance models for wire weaves. *Filtration and Separation*, **10**, 328-330.

Szabó, B.A., 1979. Some recent developments in finite element analysis. *Comput. Math. Appl.*, **5**, 99-115.

Szabó, B.A., 1986. Mesh design for the *p*-version of the finite element method. *Comput. Methods. Appl. Mech. Eng.*, **55**, 181-197.

Tripathi, A. and Chhabra, R. P., 1992. Slow power law fluid flow relative to an array of infinite cylinders. *Industrial and Engineering Chemistry Research*, **31**, 2754-2759.

Tripathi, A. and Chhabra, R. P., 1996. Transverse laminar flow of non-Newtonian fluids over a bank of cylinders. *Chemical Engineering Communication*, **147**, 197-212.

Tung, K., Shiau, J., Chuang, C., Li, Y. and Lu, W., 2002. CFD analysis on fluid flow through multifilament woven filter cloths. *Separation Science and Technology*, **37**(4), 799-821.

Turner, M. J., Clough, R. W., Martin, H. C. and Topp, L. J., 1956. Stiffness and deflection analysis of complex structures. *Journal of Aeronatutical Science*, **23**, 805-823.

Tworzydlo, W. W., Oden, J. T. and Thornton, E. A., 1992. Adaptive implicit/explicit finite element method for compressible viscous flows. *Comput. Methods Appl. Mech. Eng.*, **95**, 397-440.

Vanderplassts, G. N., 1999. *Numerical Optimization Techniques for Engineering Design*, 3rd ed., Vanderplaats Research and Development, Inc., Colorado Springs, CO, USA.

Vijaysri, M., Chhabra, R. P. and Eswaran, V., 1999. Power law fluid flow across an array of infinite circular cylinders: A numerical study. *Journal of Non-Newtonian Fluid Mechanics*, **87**, 263-282.

Wakeman, R. J. and Tarleton, E. S., 1999. Filtration: Equipment selection modelling and process simulation, Elsevier, UK.

Weighardt, K. E. G., 1953. On the resistance of screens. *Aeronautical Quarterly*, **4**, 186-192.

Wu, Y. S. and Pruess, K., 1996. Flow of non-Newtonian fluids in porous media. In *M. Y. Corapcioglu (Ed.), Advances in porous media,* **3**, Amsterdam: Elsevier, 87-178.

Zienkiewicz, O. C. and Cheung, Y. K., 1967. The finite element method in structural and continuum mechanics, McGraw-Hill, New York.

Zienkiewicz, O. C. and Taylor, R. L., 1991. The finite element method, McGraw Hill, Vol. I.

Zienkiewicz, O. C. and Taylor, R. L., 2000. The finite element method. Vol 1, Butterworth-Heinemann, Oxford.

Zienkiewicz, O. C., Gago, J. P. de S. R., and Kelly, D. W., 1983. The hierarchical concept in finite element analysis. *Comput. Struct.*, **16**(1-4), 53-65.

Zisis, Th. and Mitsoulis, E., 2002. Viscoplastic flow around a cylinder kept between parallel plates. *J. Non-Newtonian Fluid Mech.*, **105**, 1-20.

APPENDIX 1

Relevant Mathematical Bookwork

Iso-parametric Mapping

Fitting a curved boundary with straight-sided basic elements would be a less than satisfactory representation. Ideally curve-sided elements should be formulated for the task as it permits the use of a smaller number of larger elements and still achieve a close boundary representation. This is essential for three-dimensional modelling as the great number of degrees of freedom may overwhelm even the largest capacity computer. The idea underlying the development of iso-parametric element focused on mapping simple geometric shapes in some local coordinate systems into distorted shapes in the global Cartesian coordinate systems and then evaluating the resulting element equations. This advantage of finite element method has been associated with the capability to handle geometrically complex domains.

In iso-parametric mapping, a regular element called the 'master element' is selected and a local finite element approximation cased on the shape functions of this element is established. The master element is subsequently mapped into the global coordinate to generate the distorted elements. The shape functions used in the mapping functions are identical to the shape functions used to obtain finite element approximation. Finite element approximation of unknown functions in terms of locally defined shape functions can be written as

$$\bar{f} = \sum_{i=1}^{8} N_i(\xi,\eta,\zeta) f_i \tag{A1.1}$$

with f_i are nodal degrees of freedom. Consider an eight-noded linear rectangular prism element as shown in Figure A1 defined in local Cartesian coordinate system with its origin located at the centre of the element

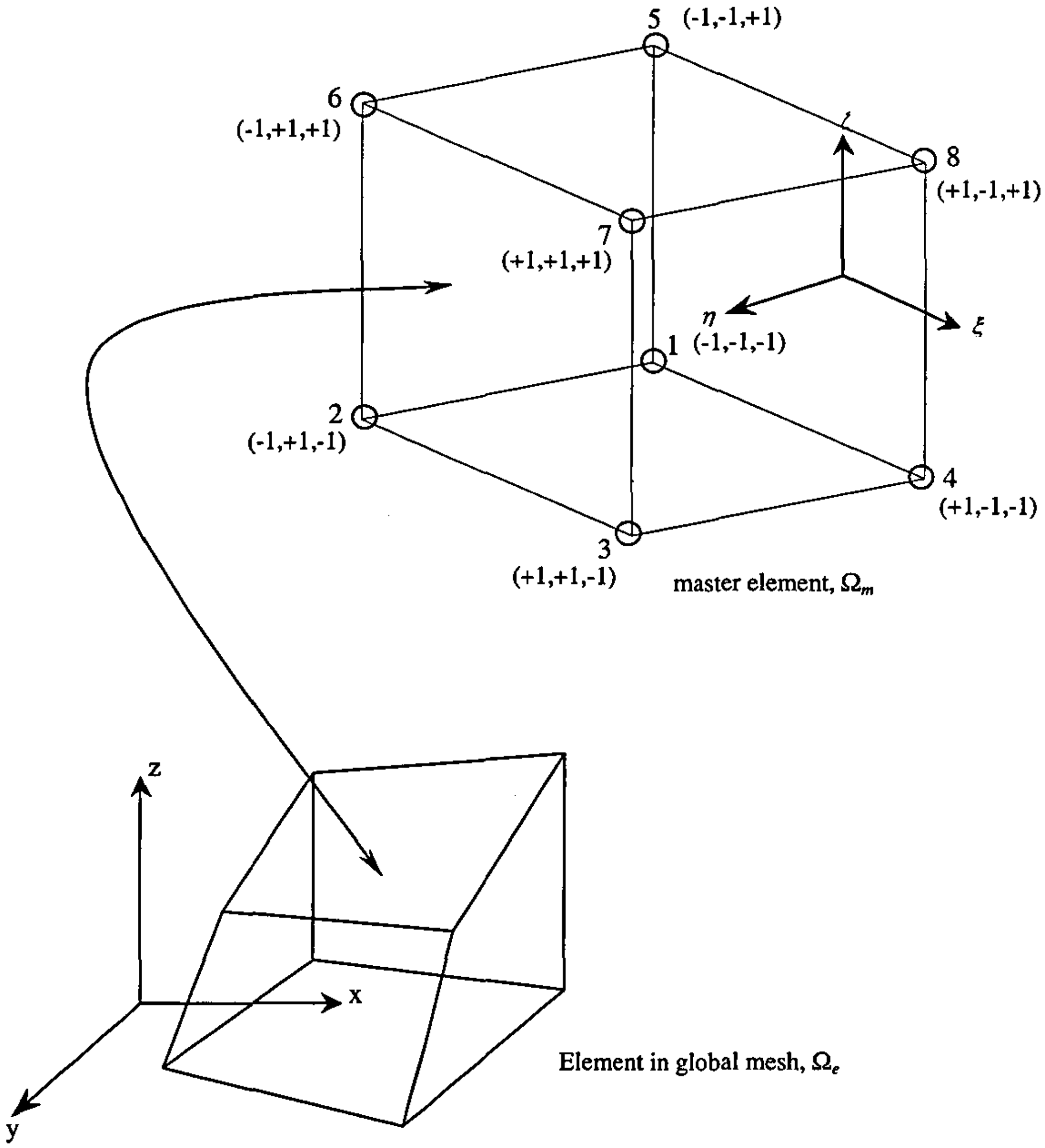


Figure A1: The mapping between a master element and an element in a global mesh.

The interpolation function for the linear prism element is given as (Huebner, 2001)

$$N_i = \frac{1}{8}(1+\xi\xi_i)(1+\eta\eta_i)(1+\zeta\zeta_i) \tag{A1.2}$$

The shape functions at each respective point shown in Figure A1 can be shown to be

$$N_1 = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta) \qquad N_2 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta) \qquad \text{(A1.3 a, b)}$$

$$N_3 = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta) \qquad N_4 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta) \qquad \text{(A1.3 c, d)}$$

$$N_5 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta) \qquad N_6 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta) \qquad \text{(A1.3 e, f)}$$

$$N_7 = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta) \qquad N_8 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta) \qquad \text{(A1.3 g, h)}$$

Consider the set of coordinates ξ, η, ζ and a corresponding set of global coordinates x, y, z. Applying the rules of partial differentiation, the derivatives for ξ, η, ζas

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi} \qquad \text{(A1.4 a)}$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \eta} \qquad \text{(A1.4 b)}$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \zeta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \zeta} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \zeta} \qquad \text{(A1.4 c)}$$

In matrix form, Eqs.(A1.4 a), (A1.4 b) and (A1.4 c) can be written as

$$\begin{Bmatrix} \dfrac{\partial N_i}{\partial \xi} \\ \dfrac{\partial N_i}{\partial \eta} \\ \dfrac{\partial N_i}{\partial \zeta} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} & \dfrac{\partial z}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial z}{\partial \eta} \\ \dfrac{\partial x}{\partial \zeta} & \dfrac{\partial y}{\partial \zeta} & \dfrac{\partial z}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial N_i}{\partial x} \\ \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial z} \end{Bmatrix} \qquad \text{(A1.5)}$$

The square matrix of the above equation is known as the Jacobian matrix where

$$J = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial y}{\partial \xi} & \dfrac{\partial z}{\partial \xi} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} & \dfrac{\partial z}{\partial \eta} \\ \dfrac{\partial x}{\partial \zeta} & \dfrac{\partial y}{\partial \zeta} & \dfrac{\partial z}{\partial \zeta} \end{bmatrix} \qquad \text{(A1.6)}$$

From the approximation

$$x(\xi,\eta,\zeta)=\sum_{i=1}^{8}N_i(\xi,\eta,\zeta)x_i \tag{A1.7 a}$$

$$y(\xi,\eta,\zeta)=\sum_{i=1}^{8}N_i(\xi,\eta,\zeta)y_i \tag{A1.7 b}$$

$$z(\xi,\eta,\zeta)=\sum_{i=1}^{8}N_i(\xi,\eta,\zeta)z_i \tag{A1.7 c}$$

Global derivatives of functions can now be related to the locally defined finite element approximation as

$$[J(\xi,\eta,\zeta)]=\begin{bmatrix} \sum_{i=1}^{8}\frac{\partial N_i}{\partial \xi}(\xi,\eta,\zeta)x_i & \sum_{i=1}^{8}\frac{\partial N_i}{\partial \xi}(\xi,\eta,\zeta)y_i & \sum_{i=1}^{8}\frac{\partial N_i}{\partial \xi}(\xi,\eta,\zeta)z_i \\ \sum_{i=1}^{8}\frac{\partial N_i}{\partial \eta}(\xi,\eta,\zeta)x_i & \sum_{i=1}^{8}\frac{\partial N_i}{\partial \eta}(\xi,\eta,\zeta)y_i & \sum_{i=1}^{8}\frac{\partial N_i}{\partial \eta}(\xi,\eta,\zeta)z_i \\ \sum_{i=1}^{8}\frac{\partial N_i}{\partial \zeta}(\xi,\eta,\zeta)x_i & \sum_{i=1}^{8}\frac{\partial N_i}{\partial \zeta}(\xi,\eta,\zeta)y_i & \sum_{i=1}^{8}\frac{\partial N_i}{\partial \zeta}(\xi,\eta,\zeta)z_i \end{bmatrix}$$

(A1.8)

with i = 1, 2, ...r. To determine the global derivatives, the Jacobian matrix is inverted and is written as

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = J^{-1}\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{Bmatrix} \tag{A1.9}$$

With this equation, expression for $\frac{\partial \bar{f}}{\partial x}$, $\frac{\partial \bar{f}}{\partial y}$ and $\frac{\partial \bar{f}}{\partial z}$ can be found directly with

$$\begin{Bmatrix} \frac{\partial \bar{f}_i}{\partial x} \\ \frac{\partial \bar{f}_i}{\partial y} \\ \frac{\partial \bar{f}_i}{\partial z} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial x}\cdots & \frac{\partial N_r}{\partial x} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_2}{\partial y}\cdots & \frac{\partial N_r}{\partial y} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z}\cdots & \frac{\partial N_r}{\partial z} \end{bmatrix}\begin{Bmatrix} \bar{f}_1 \\ \bar{f}_2 \\ \vdots \\ \bar{f}_r \end{Bmatrix} \tag{A1.10}$$

From Eq. (A1.9),

$$\begin{Bmatrix} \dfrac{\partial \bar{f}_i}{\partial x} \\ \dfrac{\partial \bar{f}_i}{\partial y} \\ \dfrac{\partial \bar{f}_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi} & \dfrac{\partial N_2}{\partial \xi} \cdots & \dfrac{\partial N_r}{\partial \xi} \\ \dfrac{\partial N_1}{\partial \eta} & \dfrac{\partial N_2}{\partial \eta} \cdots & \dfrac{\partial N_r}{\partial \eta} \\ \dfrac{\partial N_1}{\partial \zeta} & \dfrac{\partial N_2}{\partial \zeta} \cdots & \dfrac{\partial N_r}{\partial \zeta} \end{bmatrix} \begin{Bmatrix} \bar{f}_1 \\ \bar{f}_2 \\ : \\ \bar{f}_r \end{Bmatrix} \tag{A1.11}$$

To complete the evaluation of the integral, the element of volume *dxdydz* in terms of $d\xi d\eta d\zeta$

$$dxdydz = |J| d\xi d\eta d\zeta \tag{A1.12}$$

where $|J|$ is the determinant of Jacobian matrix. The operations indicated in Eq. (A1.9) and Eq. (A1.12) depend on the existence of $[J]^{-1}$ for each element of the assembly, and the coordinate mapping described by Eq. (A1.7 a-c) is unique only if $[J]^{-1}$ exists.

With these transformations, the integral such as

$$\int_{V_e} f\left(\phi, \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z}\right) dxdydz \tag{A1.13}$$

with V_e the volume of distorted element in the *x-y-z* coordinate system, is reduced to

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f'(\phi, \xi, \eta, \zeta) d\xi d\eta d\zeta \tag{A1.14}$$

with f' the transformed function f. This integral is subsequently solved by numerical integration using the Gauss-Legendre quadrature. The details of quadrature technique, sampling points and weighting factors can be referred at Zienkiewicz and Taylor (2000).

Partial derivatives with respect to ξ

$$\frac{\partial N_1}{\partial \xi} = -\frac{1}{8}(1-\eta)(1-\zeta) \qquad \frac{\partial N_2}{\partial \xi} = -\frac{1}{8}(1+\eta)(1-\zeta) \qquad \text{(A1.15a, b)}$$

$$\frac{\partial N_3}{\partial \xi} = \frac{1}{8}(1+\eta)(1-\zeta) \qquad \frac{\partial N_4}{\partial \xi} = \frac{1}{8}(1-\eta)(1-\zeta) \qquad \text{(A1.15c, d)}$$

$$\frac{\partial N_5}{\partial \xi} = -\frac{1}{8}(1-\eta)(1+\zeta) \qquad \frac{\partial N_6}{\partial \xi} = -\frac{1}{8}(1+\eta)(1+\zeta) \qquad \text{(A1.15e, f)}$$

$$\frac{\partial N_7}{\partial \xi} = \frac{1}{8}(1+\eta)(1+\zeta) \qquad \frac{\partial N_8}{\partial \xi} = \frac{1}{8}(1-\eta)(1+\zeta) \qquad \text{(A1.15g, h)}$$

Partial derivatives with respect to η

$$\frac{\partial N_1}{\partial \eta} = -\frac{1}{8}(1-\xi)(1-\zeta) \qquad \frac{\partial N_2}{\partial \eta} = \frac{1}{8}(1-\xi)(1-\zeta) \qquad \text{(A1.16a, b)}$$

$$\frac{\partial N_3}{\partial \eta} = \frac{1}{8}(1+\xi)(1-\zeta) \qquad \frac{\partial N_4}{\partial \eta} = -\frac{1}{8}(1+\xi)(1-\zeta) \qquad \text{(A1.16c, d)}$$

$$\frac{\partial N_5}{\partial \eta} = -\frac{1}{8}(1-\xi)(1+\zeta) \qquad \frac{\partial N_6}{\partial \eta} = \frac{1}{8}(1-\xi)(1+\zeta) \qquad \text{(A1.16e, f)}$$

$$\frac{\partial N_7}{\partial \eta} = \frac{1}{8}(1+\xi)(1+\zeta) \qquad \frac{\partial N_8}{\partial \eta} = -\frac{1}{8}(1+\xi)(1+\zeta) \qquad \text{(A1.16g, h)}$$

Partial derivatives with respect to ζ

$$\frac{\partial N_1}{\partial \eta} = -\frac{1}{8}(1-\xi)(1-\eta) \qquad \frac{\partial N_2}{\partial \eta} = -\frac{1}{8}(1-\xi)(1+\eta) \qquad \text{(A1.17a, b)}$$

$$\frac{\partial N_3}{\partial \eta} = -\frac{1}{8}(1+\xi)(1+\eta) \qquad \frac{\partial N_4}{\partial \eta} = -\frac{1}{8}(1+\xi)(1-\eta) \qquad \text{(A1.17c, d)}$$

$$\frac{\partial N_5}{\partial \eta} = \frac{1}{8}(1-\xi)(1-\eta) \qquad \frac{\partial N_6}{\partial \eta} = \frac{1}{8}(1-\xi)(1+\eta) \qquad \text{(A1.17e, f)}$$

$$\frac{\partial N_7}{\partial \eta} = \frac{1}{8}(1+\xi)(1+\eta) \qquad \frac{\partial N_8}{\partial \eta} = \frac{1}{8}(1+\xi)(1-\eta) \qquad \text{(A1.17g, h)}$$

APPENDIX 2

Publications

(1) FILTRATION SOCIETY CONFERENCE ON "FILTER MEDIA FOR GAS/LIQUID APPLICATIONS"

May 10, 2005, Runcorn, United Kingdom

Title: Modelling Flow in Monofilament Cloths – Prediction of Pressure Loss

Status: Oral presentation.
Presented and published in conference proceedings

Citation:

K.C. Ting, R.J. Wakeman and V. Nassehi, 2005. Modelling of Modelling Flow in Monofilament Filter Cloths – Prediction of Pressure Loss, *Filtration Society. Conference*, Runcorn, United Kingdom.

Modelling Flow in Monofilament Filter Cloths – Prediction of Pressure Loss

K.C. Ting, R.J. Wakeman and V. Nassehi
Advanced Separation Technologies Group,
Department of Chemical Engineering, Loughborough University,
Loughborough, LE11 3TU, UK.

ABSTRACT

Monofilament filter cloths are used as the separation media in filtration; woven wire cloths or screens are also used as the media in filters or to enhance the integrity of the filter medium in, for example, filter cartridges. In this paper we present research results aimed at simulating non-Newtonian fluid flow through a woven cloth. Due to the complex geometry of a woven cloth, 3-D modelling is necessary to correctly visualise the structure of the flow and hence to predict pressure losses. The modelling in a 3-D domain was handled using a finite element method which is known to cope with flow domains in complex geometries very effectively. The governing equations of continuity and momentum were solved by a mixed U-V-W-P finite element method and in conjunction with a first order Taylor-Galerkin scheme for temporal discretization. The simulation results were found to be in good agreement with experimental data, showing the developed model is capable of generating accurate results for flow of both Newtonian and non-Newtonian fluids through filter media.

Keywords: Woven media, wire cloths, wire mesh screens, finite element method, three-dimensional, Newtonian fluids, non-Newtonian fluids, pressure loss.

INTRODUCTION

Whilst woven fabrics are commonly used as filter media, metal and plastic wire screens have also been widely used in the construction of filters and other separators, either as the support for finer filter media or as the filtering medium itself. However, the presence of a mesh changes the characteristics of the fluid flow and can affect the filtering effectiveness of the medium. Many studies have been conducted to investigate fluid flow past ideal shapes such as cylinders and spheres for Newtonian fluids, but there are relatively few studies of flow through woven fabrics and wire screens.

Fluid flow through cloths and screens has been studied experimentally by various researchers (Wieghardt, 1953; Armour and Connon, 1968; Rushton, 1969; Rushton and Griffiths, 1971; Ehrhardt, 1983; Squiers, 1984; Chhabra and Richardson, 1985; Kiljanski and Dziubinski, 1996). Most of these researchers used various gases and Newtonian liquids as the experimental fluids in their work. For instance, Rushton (1969) presented a comprehensive series of experimental data on flow of air and water through filter cloths, while Rushton and Griffiths (1971) presented various approaches to explaining the flow situation in woven fabrics – those approaches included an orifice analogy, a randomly packed bed analogy, and the analytical solution for creeping flow over cylinders.

Investigation of flow of non-Newtonian liquids through single screens are very limited. Chhabra and Richardson (1985) studied experimentally flow through a screen using a Newtonian liquid and shear-thinning carboxymethyl cellulose (CMC) solutions of various concentration with n (the Power Law index) between 0.34 and 0.61. The correlation between the drag coefficient and the Reynolds number (Re_{NN}) was investigated and reported. Kiljanski and Dziubinski (1996) furthered the study to include sets of filter screens using shear thinning molten polyethylene for a lower range of *Re* and provided correlation of drag coefficient with *Re* for multiple screens. None of these researchers studied the velocity profile when the fluids flow through the screen.

Lu *et al.* (1996) lamented that most previous studies on fluid flow through woven structures have focused on the pressure drop problem and not the macroscopic details of the velocity profile and pressure contour in the weave apertures. They argued that a better understanding of the flow pattern in the woven structure could prove useful in examining the initial stages of cake filtration as well as the effect of weaves on fouling phenomena within a filter cloth. They studied the influence of fabric pore construction on the flow pattern in the interstices and downstream of a cloth using a commercial CFD software FLUENT^{TM}, and discovered that the flow pattern in the interstices and downstream were different for each basic pore type. As the position of the upper filament in each pore type differs, the flow of water which swirls around the upper surfaces of the upper filament into the narrow channel between the cloth filaments was characteristically unique for each case. Tung *et al.* (2002) furthered the study to include fluid flow through multifilament cloths and spun staple yarn. By assigning different permeability values to the woven filament, they investigated the distribution of fluid flow in interyarn and interfibre pores of four basic pore types. Their simulations results showed a similar pattern of pressure drop to that of Lu *et al.* where pore type 1 were found to give the highest pressure drop and pore type 4 gives the lowest pressure drop.

Non-Newtonian flow processes characterize many polymer engineering operations, and very little is known about the boundary layer flows for non-Newtonian fluids. An examination of the available literature revealed that most of the boundary layer literature pertains to the simple power-law fluid model and to the simple shapes including flat plates, spheres and cylinders (Skelland, 1967; Astarita and Marrucci, 1974; Schowalter, 1978; Chhabra, 1999; Chhabra and Richardson, 1999). Considering the significance of knowledge in the flow behaviour of non-Newtonian fluids, it is necessary to research the complex flow field generated by the flow of non-Newtonian fluid through a complex geometry. Hence, the significance of studying the flow of a non-Newtonian fluid through a wire mesh cannot be over-stressed.

Computational fluid dynamics (CFD) techniques provide a powerful and convenient route for the qualitative and quantitative analysis of the non-Newtonian flow systems. With increasingly powerful processors and more economical computing cost, accurate computations of very complex flow problems have been made possible in 2-D and even 3-D domains. Therefore, CFD seems to be a feasible alternative to study flow through weave structures that are characterized by small apertures in the mm and μm size ranges.

MATHEMATICAL MODEL

A 3-D mathematical model based on flow and constitutive equations has been developed. The majority of highly viscous non-Newtonian fluid flows are characterized as low Reynolds number Stokes flow regimes in which the inertia terms in the equation of motion are neglected. In addition, highly viscous flow systems are in general dominated by stress and pressure variations and the body forces acting on the fluid are relatively small and can be justifiably ignored. The governing Stokes flow equations in Cartesian coordinate systems are summarised below.

Equation of Continuity

The continuity equation for an incompressible fluid is written as $\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0$. However, a slightly perturbed form of this equation is used here and the equation is written as:

$$\frac{1}{\rho c_s^{\,2}}\frac{\partial p}{\partial t}+\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \qquad (1)$$

where c_s, p, v, ρ represent the speed of sound in the fluid, pressure, velocity and the fluid density, respectively.

This perturbed form of the continuity equation has been used in order to satisfy the Ladyzhenskaya-Babuska-Brezzi stability condition (Ladyzhenskaya, 1969; Babuška, 1971; Brezzi, 1974). The use of this continuity equation, which corresponds to slightly compressible fluids, allows the utilization of an equal order interpolation model for the velocity and pressure, and hence increases the flexibility of the developed solution scheme (Nassehi, 2002), and has been applied successfully to modelling flows in pleated cartridge filters (Ruziwa *et al.*, 2004; Hanspal *et al.*, 2004; Nassehi *et al.*, 2005).

Momentum Equation

The momentum equation can be written as

$$\rho\frac{\partial v_i}{\partial t}=-\frac{\partial p}{\partial i}+\frac{\partial}{\partial i}\left[2\eta\frac{\partial v_i}{\partial i}\right]+\sum\frac{\partial}{\partial i}\left[\eta\left(\frac{\partial v_i}{\partial j}+\frac{\partial v_j}{\partial i}\right)\right] \qquad (2)$$

where $i=x, y, z$ and $j\neq i$, and η is fluid viscosity.

Constitutive Equation

The viscosity of the non-Newtonian fluid is updated in the solution algorithm using the power law model:

$$\eta=\eta_0\left(\dot{\gamma}\right)^{n-1} \qquad (3)$$

where η is the apparent viscosity, η_0 is the consistency index, n is the power law index and $\dot{\gamma}$ is the shear rate.

Numerical Scheme

The numerical scheme used in this work is the *Mixed Finite Element* U-V-W-P technique (Zienkiewicz and Taylor, 2000) in which both velocity and pressure in the governing equations are regarded as primary variables and are discretized as unknowns. The first order Taylor-Galerkin method was used to discretize the equations.

MODEL VALIDATION

Computational Mesh

3-D test cases were devised to check the computer codes developed. Three test cases were designed in increasing mesh complexity to progressively test the capability of the algorithm.

Test Case 1 was used to validate the capability of the program in solving the continuity and motion equations. The main test for the accuracy of the simulation results is ensuring a mass balance over the domain under investigation. A rectangular domain, shown in Figure 1, was adapted with four impermeable, no-slip, vertical walls and the inlet was at the upper horizontal surface and the outlet at the bottom horizontal surface. The ratio of the domain length (x), width (y) and height (z) was 3:3:5.

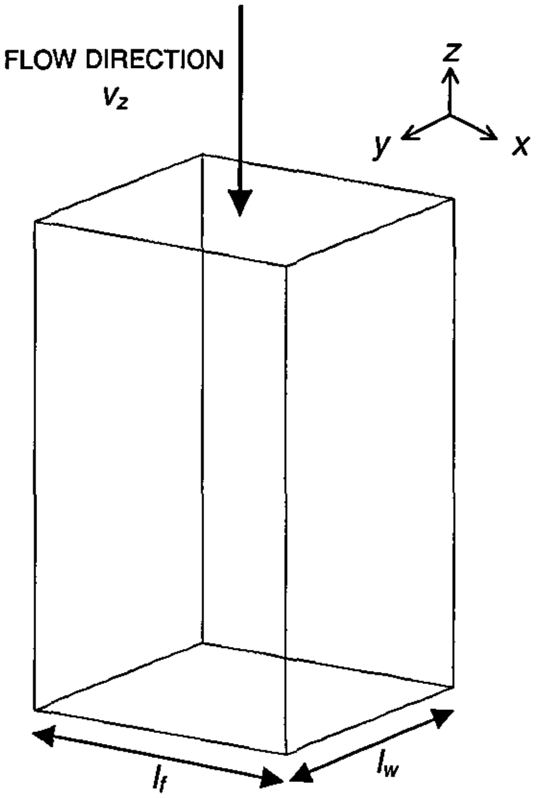


Figure 1 A schematic diagram of *Test Case 1*.

Test case 2 is slightly more complex with two half-cylinders protruding into a rectangular domain, emulating the flow around two parallel fibres that form two sides of an aperture. The ratio of aperture size to wire diameter used in this test case was 1.5 (values from about 1.1 up to about 7 are commonly used ratios in industrial wire meshes). The inlet was again the upper horizontal surface and the outlet was at the bottom horizontal surface. The half cylinder surfaces were considered to be impermeable and non-slip. The ratio of the domain length (x), width (y), height (z) and wire diameter (d) was 5:5:2:2 and the schematic computational mesh is shown in Figure 2.

Test Case 3 has all the basic features of a typical wire mesh domain except that the four supposedly intertwined weft and warp wires were flattened to a horizontal plane to test the response of the model to highly complex geometry and mesh distortion. The four cylinders were jointed together via a 45° slice at each corner. The ratio of the domain length (x), width (y), height (z) and wire diameter (d) was 5:5:2:2. The results from this test case provided a powerful pretext into understanding the complex flow field in different pore types of a wire mesh. Inlet and outlet settings were similar to *Test Case 1* and *Test Case 2*. The half cylinder surfaces were again considered to be impermeable and non-slip. The schematic domain is shown in Figure 3.

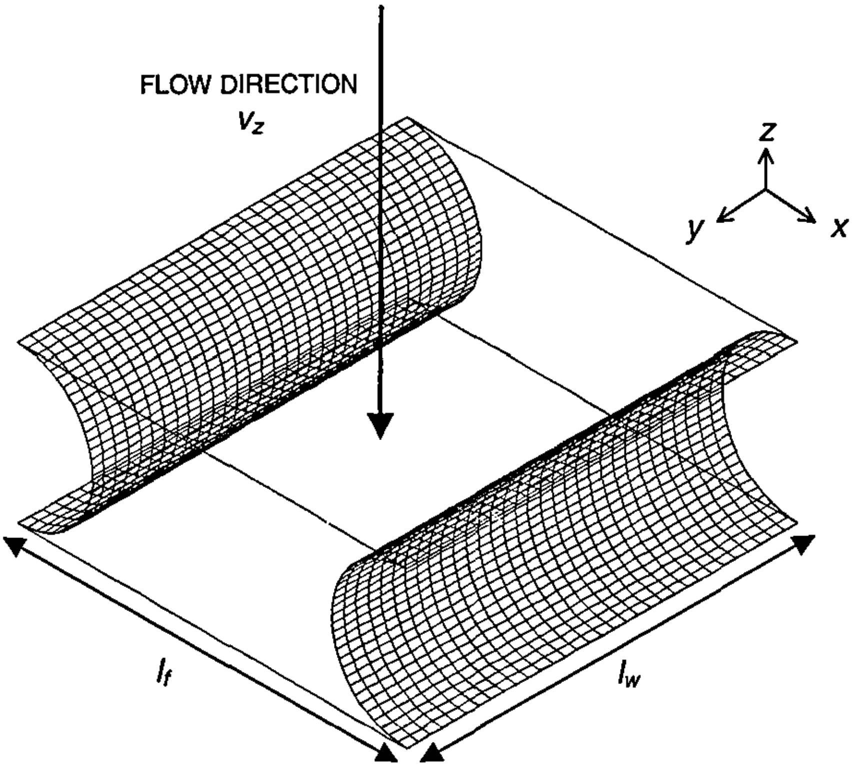


Figure 2 A schematic diagram of *Test Case 2*.

Boundary Conditions

The velocity v_z corresponds to the approach velocity of the fluid towards the aperture and was set as 0.1 m s^{-1} at the inlet. The boundaries (the cloth filaments) were considered to be impermeable, non-slip surfaces where $v_x = v_y = v_z = 0$. The computational mesh and degree of freedom (DOF) used for each test case are given in Table 1.

Table 1 Computational mesh used for each test cases.

Test Case	No. of elements	No. of nodes	DOF
1	1920	2511	3544
2	4096	4913	4572
3	4096	4913	4164

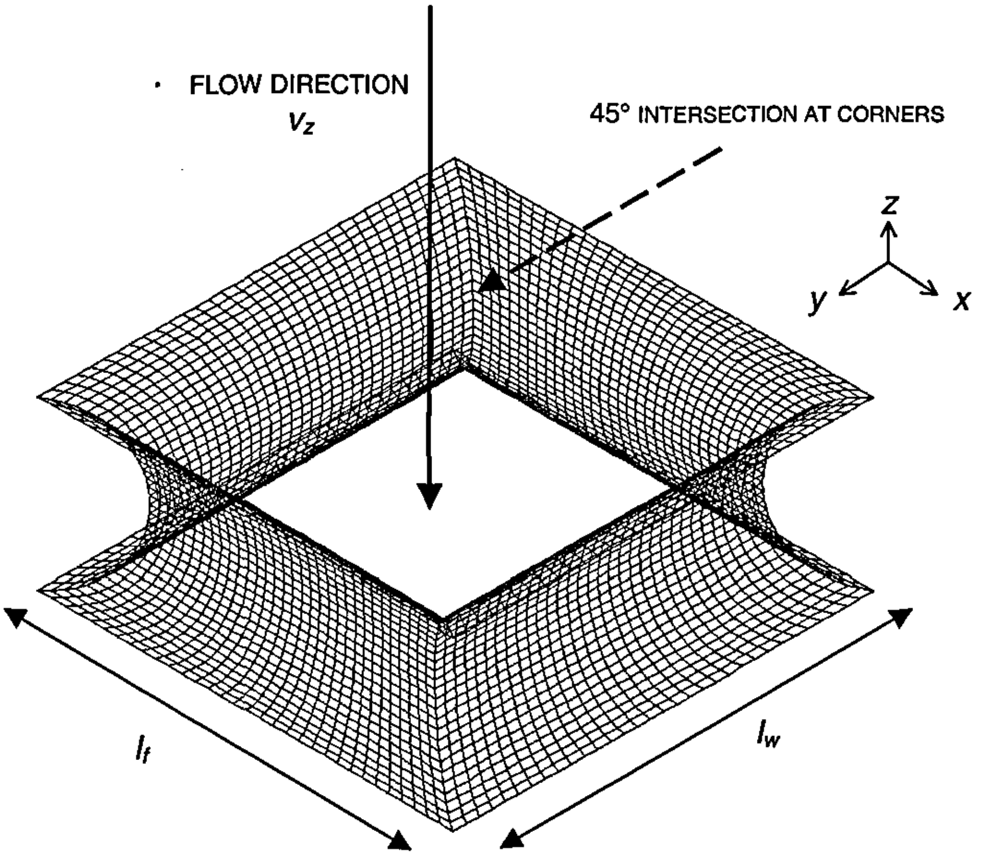


Figure 3 A schematic diagram of *Test Case 3.*

Physical Properties and Numerical Parameters

A non-Newtonian fluid with the properties given in Table 2 was used in the simulations. Similar properties were used for the simulations of the Newtonian fluid, but with n taken to be 1.0.

Table 2 Physical properties of the non-Newtonian fluid used.

Physical Property	Value
Density, ρ	970 kg m^{-3}
Consistency index, η_0	80.0 kg m^{-1} s^{-1}
Power Law index, n	0.90

A time step of $\Delta t = 0.01$ s was used throughout the simulations and the time stepping parameter θ used for the flow modelling was 0.95.

An Intel Pentium IV 2.6 GHz processor has been used to execute the computation to produce simulation results. The pre-processing work of the domain development and the mesh generation were handled using COSMOS GEOSTAR. Post-processing presentation of field variables such as flow velocity vectors and pressure contours were plotted using the commercial graphic software package SURFER 8.

VALIDATION RESULTS

Figure 4 illustrates a typical velocity vectors of *Test Case 1* in the *x*-*z* plane at $y = l_w/2$ for non-Newtonian fluid. The maximum *z*-direction velocity v_z of non-Newtonian fluid were found to be 4.5% higher that of the Newtonian liquid. This can be explained by the shear-thinning effect of the non-Newtonian fluid. The overall mass balance for Newtonian and non-Newtonian simulations is 99.7% and 99.4%, respectively. The excellent accuracy showed that continuity was preserved in the simulations.

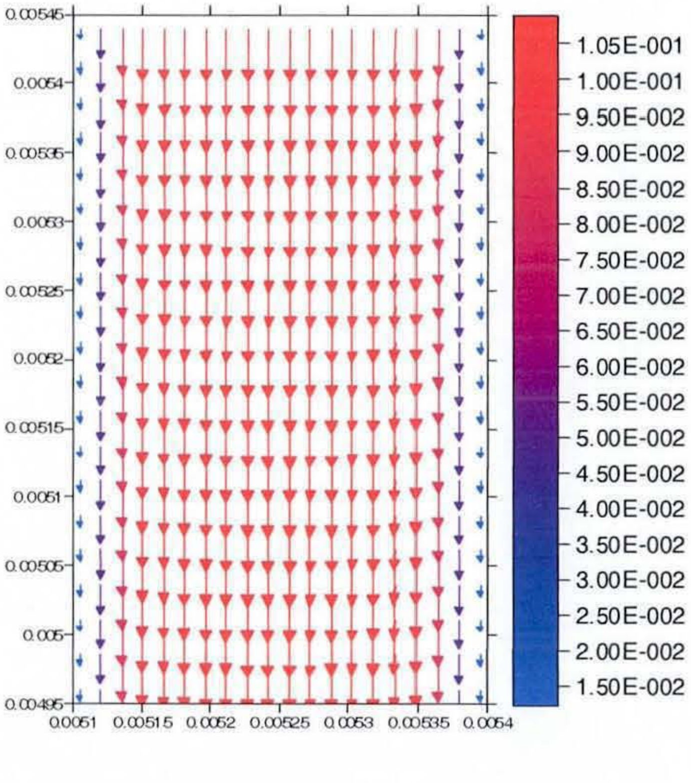
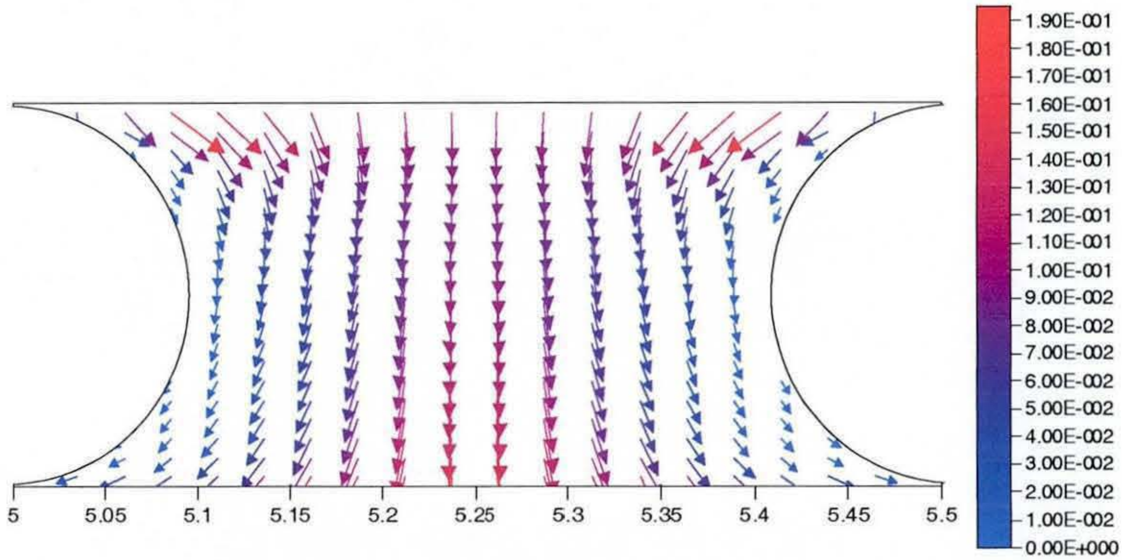
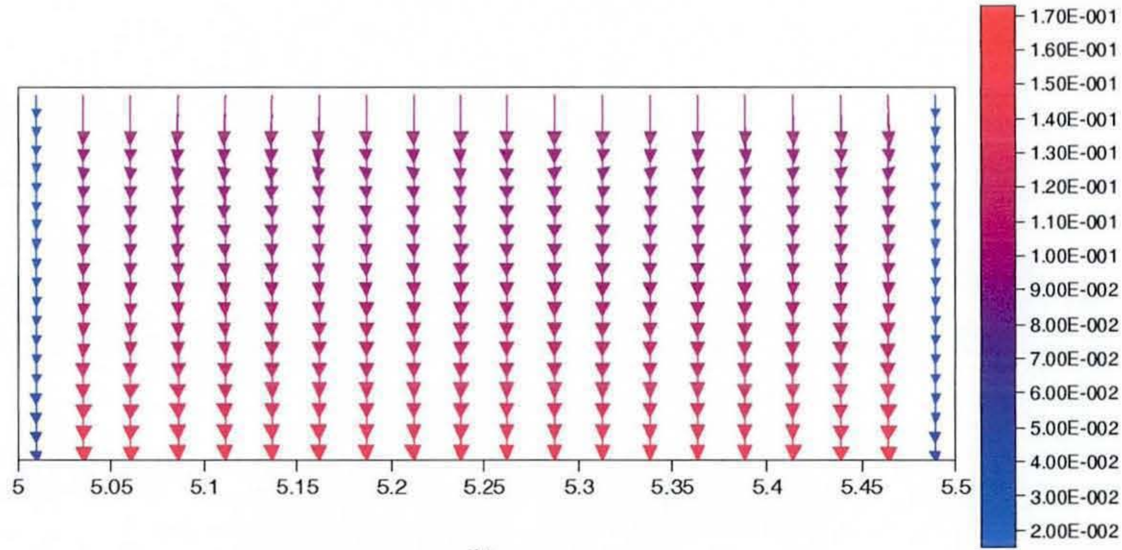


Figure 4 The velocity vectors of test case 1 in the *x*-*z* plane at $y = l_w/2$ for a non-Newtonian fluid.

Figure 5 shows a typical velocity vectors of *Test Case 2* for a non-Newtonian fluid in the x-z plane at $y = l_w/2$ and in the y-z plane at $x = l_f/2$ Higher velocities were observed at the area immediately above the cylinders and at the section between the cylinders compared to the velocity at the inlet and outlet. This is attributed to the compression of the fluid that leads to the acceleration of the fluid velocity. The overall mass balance for Newtonian and non-Newtonian simulations in *Test Case 2* showed an outlet flow of 95.0% and 94.8% compared to the inlet flow. The discrepancy is due to the relatively coarse mesh used and can be improved by further mesh refinement. This is evident when an earlier run with a very coarse mesh of 512 elements gave a mass balance of 86.2% for Newtonian liquid.



(a)



(b)

Figure 5 The velocity vectors of test case 2 in the (a) x-z plane at $y = l_w/2$ (b) y-z plane at $x = l_f/2$ for non-Newtonian fluid.

Figure 6 depicted the a typical simulation results of *Test Case 3* for a non-Newtonian fluid in the *x-z* plane at $y = l_w/2$ and in the *y-z* plane at $x = l_f/2$ The highly complex geometry of the four adjoined cylinders formed a narrow channel for the fluid to flow through. The mass balance for Newtonian and non-Newtonian simulations is 96.1% and 95.8%, respectively. Simulations using a very coarse mesh of 512 elements gave a mass balance of 86.3%. These results shows that the mass balance accuracy can be improved by mesh refinement. The overall consistency in the results showed that the model is capable of solving 3-D flow problem in a domain with complex geometry.

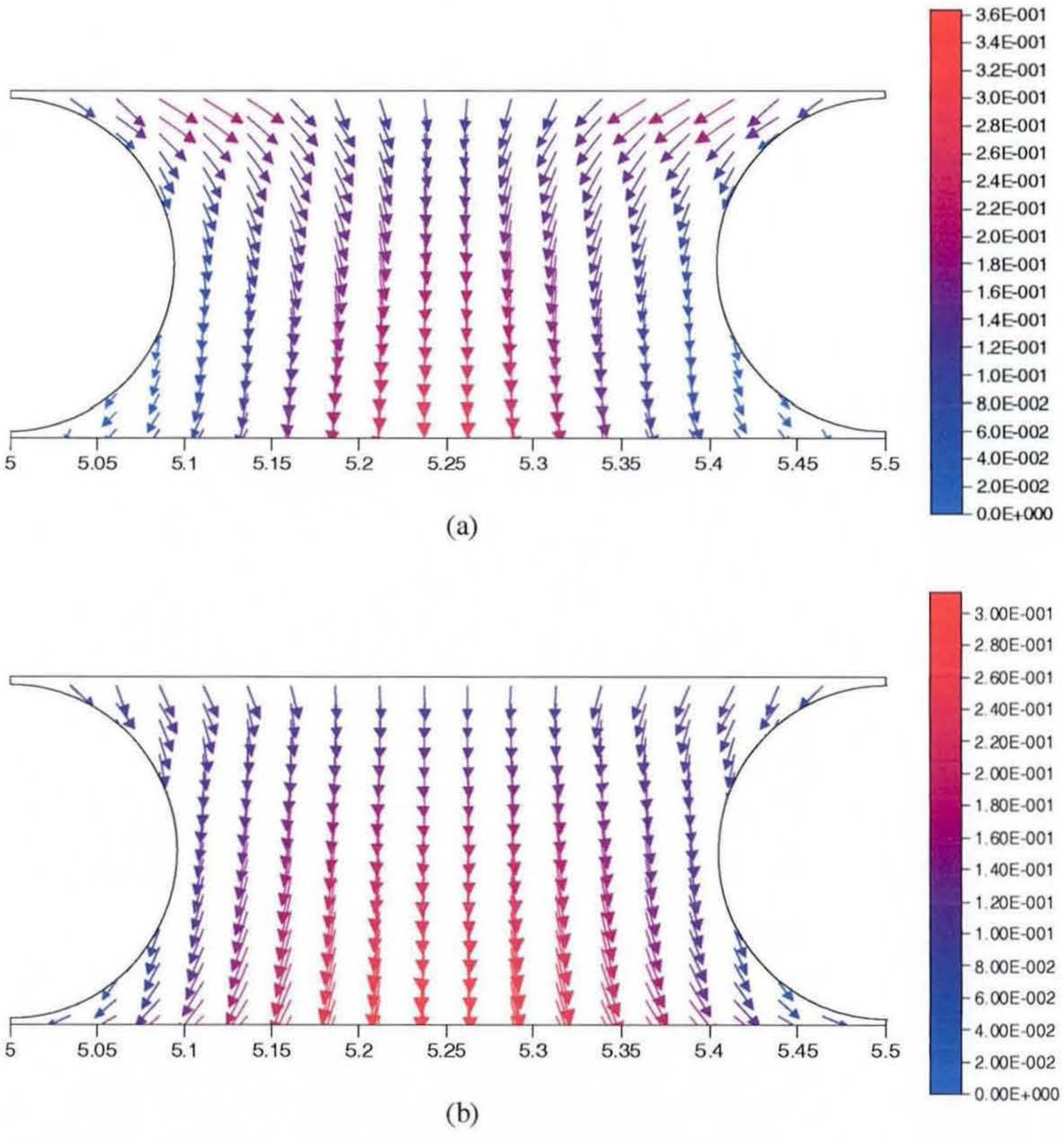
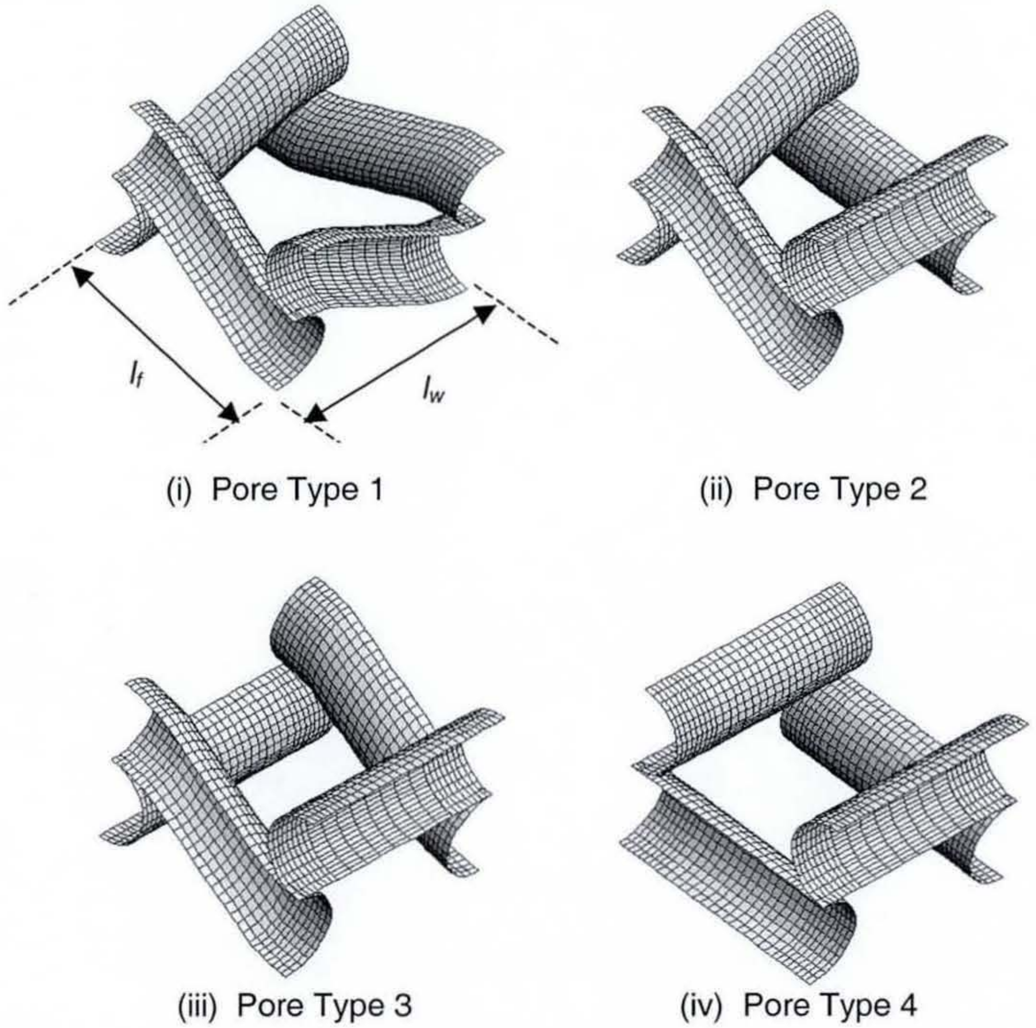


Figure 6 The velocity vectors of test case 3 in the (a) *x-z* plane at $y = l_w/2$
(b) *y-z* plane at $x = l_f/2$ for non-Newtonian fluid.

WOVEN MEDIA

The model can now be applied to the simulation of flow through a monofilament woven medium. The weaves considered in this work are the plain weave, the twill weave, and the satin weave. These give rise to the four types of basic pore shape (Backer, 1951) and are shown in Figure 7, where only the half circle of each filament is illustrated.



(i) Pore Type 1

(ii) Pore Type 2

(iii) Pore Type 3

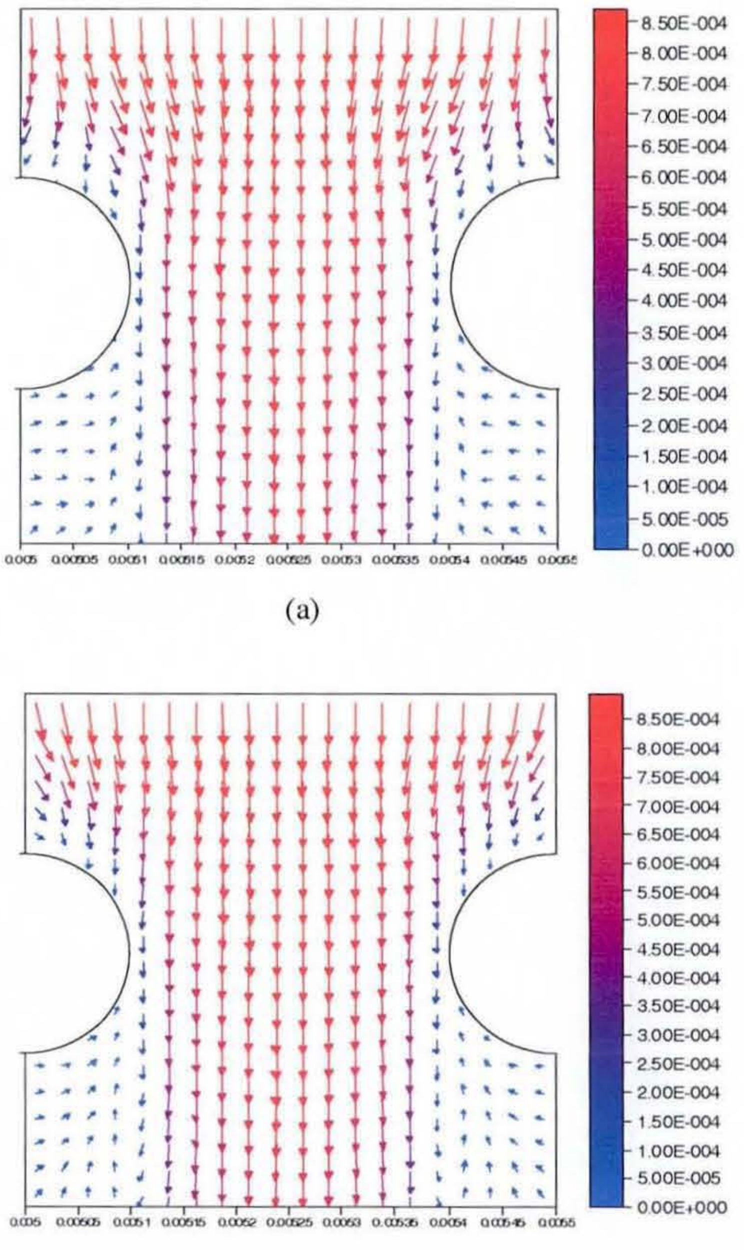
(iv) Pore Type 4

Figure 7 Four types of basic weave patterns of a wire mesh.

RESULTS AND DISCUSSIONS

Fluid flow through a plain weave (pore type 1, Figure 7) was first investigated in conjunction with the experimental studies by Rushton (1969) using water and Chhabra and Richardson (1985) using shear thinning fluids. Simulations were tested against experimental data extracted from both studies. The ratio of aperture size to wire diameter used is 1.5 and the ratio of the domain length (x), width (y), height (z) and wire diameter (d) is 5:5:5:2. The flow inlet is at the top horizontal surface while the flow exit is at the lower horizontal surface. A computational grid with 4608 elements and 6560 nodes with 3363 DOF was used in the simulations.

A typical velocity vectors for the flow through wire mesh is illustrated in Figure 8. The flow is observed to swirl inwards on both sidewalls towards the centre on both *x-z* and *y-z* planes. This is consistent with the location of the wire mesh on both planes. Higher velocities were also observed at the area immediately above the wire mesh and in the section between the wires, compared to the velocity at the inlet and outlet.

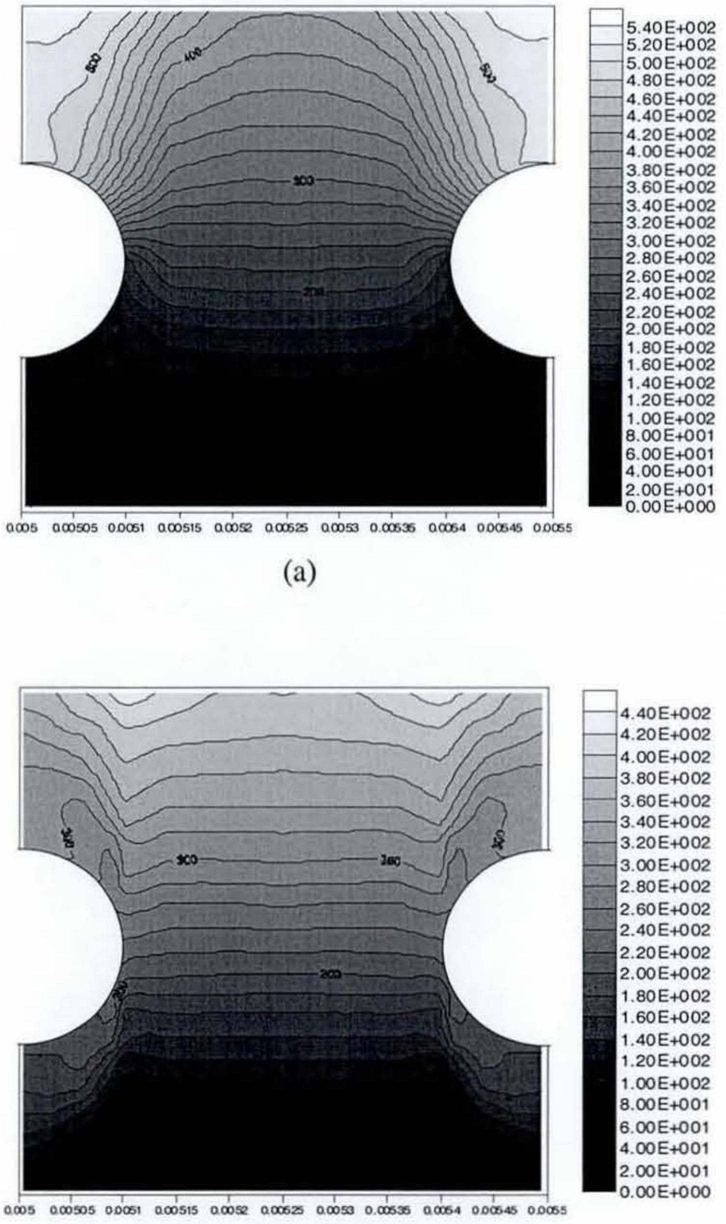


(a)

(b)

Figure 8 Velocity vectors for flow of non-Newtonian fluid through pore type 1 at (a) *x-z* plane at $y = l_w/2$ (b) *y-z* plane at $x = l_f/2$.

The pressure profile illustrated in Figure 9 shows the expected high pressure on the surface of the wire meshes and decreasing rapidly as the fluid flows through the wire meshes. Cross section plots were presented for pressure contours in both the *x*-*z* plane at $y = l_w/2$ and the *y*-*z* plane at $x = l_f/2$.



(a)

(b)

Figure 9 Pressure contour for flow of non-Newtonian fluid through pore type 1 at (a) *x*-*z* plane at $y = l_w/2$ (b) *y*-*z* plane at $x = l_f/2$.

The physical properties of the test fluids and the characteristic dimensions of the wire mesh and monofilament cloths used in Chhabra and Richardson's (1985) and Rushton's (1969) experiments and in this study are summarized in Table 3.

Table 3 Physical properties of the test fluids and characteristic dimensions of the plain weave wire mesh screens and monofilament cloth.

Test Liquid	n	η_0	Aperture	d	Symbols in
		(Pa s)	(μm)	(μm)	Figure 10
Chhabra and Richardson (1985); wire mesh screens					
1.5% CMC in water	0.60	6.6	53	36	■
1.5% CMC in water	0.59	7.6	53	36	◆
2.0% CMC in water	0.61	9.5	150	100	▲
2.5% CMC in water	0.34	45.0	150	100	●
Rushton (1969); monofilament Nylon cloth					
Water	1.0	0.001	24	30	+
	1.0	0.001	42.4	37	+
	1.0	0.001	60	30	+
	1.0	0.001	59	44	+
	1.0	0.001	71	61	+
	1.0	0.001	99	75	+
	1.0	0.001	144	105	+
	1.0	0.001	186	153	+
This study					
1.5% CMC in water	0.60	6.6	300	200	□
1.5% CMC in water	0.59	7.6	300	200	◇
2.0% CMC in water	0.61	9.5	300	200	△
2.5% CMC in water	0.34	45.0	300	200	○
Water	1.0	0.001	300	200	×

Chhabra and Richardson presented their experimental results for shear-thinning fluids using a loss coefficient C_D (which we refer to as a drag coefficient) defined as

$$C_D = \frac{2\Delta p}{\rho(u/\varepsilon)^2}\frac{1}{(1-\varepsilon)} \tag{4}$$

where ε is the porosity of the wire mesh and Δp the pressure drop across the wire mesh. They also used a modified Reynolds number for a non-Newtonian fluid to $\mathrm{Re}_{NN} = \rho(u/\varepsilon)^{2-n} d^n / \eta_0$, a form similar to that used by Metzner (1956) and Skelland (1967) with porosity taken into consideration. In their paper, however, they erroneously took d to be the wire diameter rather than the hydraulic diameter, d_h, which can be taken as the aperture side dimension in the case of a square mesh. A more appropriate correlation for Re_{NN} consistent to that used by earlier researchers is therefore given as:

$$\mathrm{Re}_{NN} = \frac{\rho\left(u/\varepsilon\right)^{2-n} d_h^{\ n}}{\eta_0} \tag{5}$$

where Re_{NN} reduces to the Reynolds number for a Newtonian liquid when $n = 1$. The experimental results presented in their paper were recalculated using equation (5) and presented in Figure 10. The results for water flow through monofilament nylon cloths reported by Rushton (1969) were also calculated using equations (4) and (5) and included on Figure 10. For the simulations in this work, the relationship between the flow rate and the pressure drop for each fluid is calculated from the model and subsequently presented using equations (4) and (5) and also plotted in Figure 11. From Figure 10, the simulation results for water and the non-Newtonian liquid were observed to be in good agreement with the experimental results. Furthermore, the drag coefficient C_D and the Reynolds number Re_{NN} are related by a best fit curve of

$$C_D = \frac{10}{\mathrm{Re}_{NN}^{\ \ 1.14}} \tag{6}$$

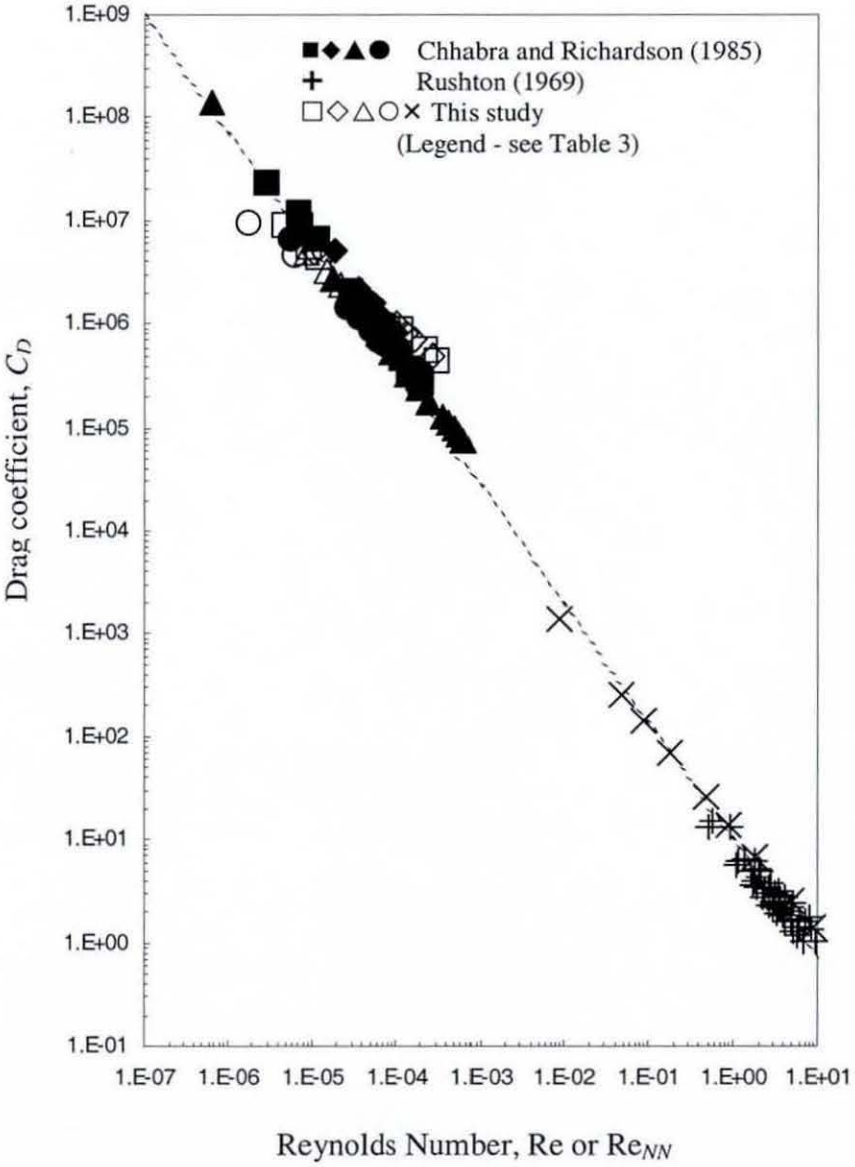


Figure 10 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian liquid flows through plain weave monofilament cloths and meshes.

The $C_D\text{Re}_{NN}{}^{1.14}$ values for the simulation results in this study are tabulated in Table 4. The simulation results obtained in this study are in good agreement to the experimental studies conducted by earlier researchers and this gives confidence to the developed model for its capability in simulating Newtonian and non-Newtonian fluid flow through a complex domain.

Table 4 $C_D\text{Re}_{NN}{}^{1.14}$ for the fluids used in this study.

Fluid	*n*	η_0	$C_D\text{Re}_{NN}{}^{1.14}$
		(Pa.s)	
Water	1.0	0.001	0.87 * 10
1.5% CMC in water	0.34	45.0	0.26 * 10
1.5% CMC in water	0.61	9.5	0.80 * 10
2.0% CMC in water	0.59	7.6	1.03 * 10
2.5% CMC in water	0.60	6.6	1.11 * 10

The simulations were extended to pore types 2, 3 and 4 (see Figure 7). The fluid properties given in Table 2 were used. The drag coefficient C_D and the Reynolds number Re_{NN} for each pore types were calculated using equations (5) and (6) and plotted in Figure 11. From Figure 11, it can be observed that pore type 1 gives the lowest pressure drop, while pore type 3 gives the highest pressure drop across the wire mesh with pore type 2, 4 in between them. This observation however is in serious disagreement to the observation by Lu *et al.* (1996) and Tung *et al.* (2002) as they reported the lowest pressure drop for pore type 4 and highest pressure drop for pore type 1. To explain this discrepancy, the experimental data of previous researchers and the industrial wire mesh manufacturers were consulted. A closer examination of the experimental results reported by Armour and Cannon (1968) and Rushton and Griffiths (1971) showed a consistently lower pressure drop for plain weave compared to twilled weave, with satin weave ranging between them. As plain weave has the most uniform pore type amongst other weave patterns, where it consists entirely of pore type 1, plain weave can be safely used as a benchmark to check the accuracy of the numerical model. The consistency of our results to that of Armour and Cannon (1968) and Rushton and Griffiths (1971) showed the accuracy of our predictions. Personal communication with the industrial experts and wire mesh manufacturer has confirmed the industrial practice whereby plain weave is primarily used due to its lowest flow resistance. These cross-examinations confirmed the validity of our results and gives confidence to our model.

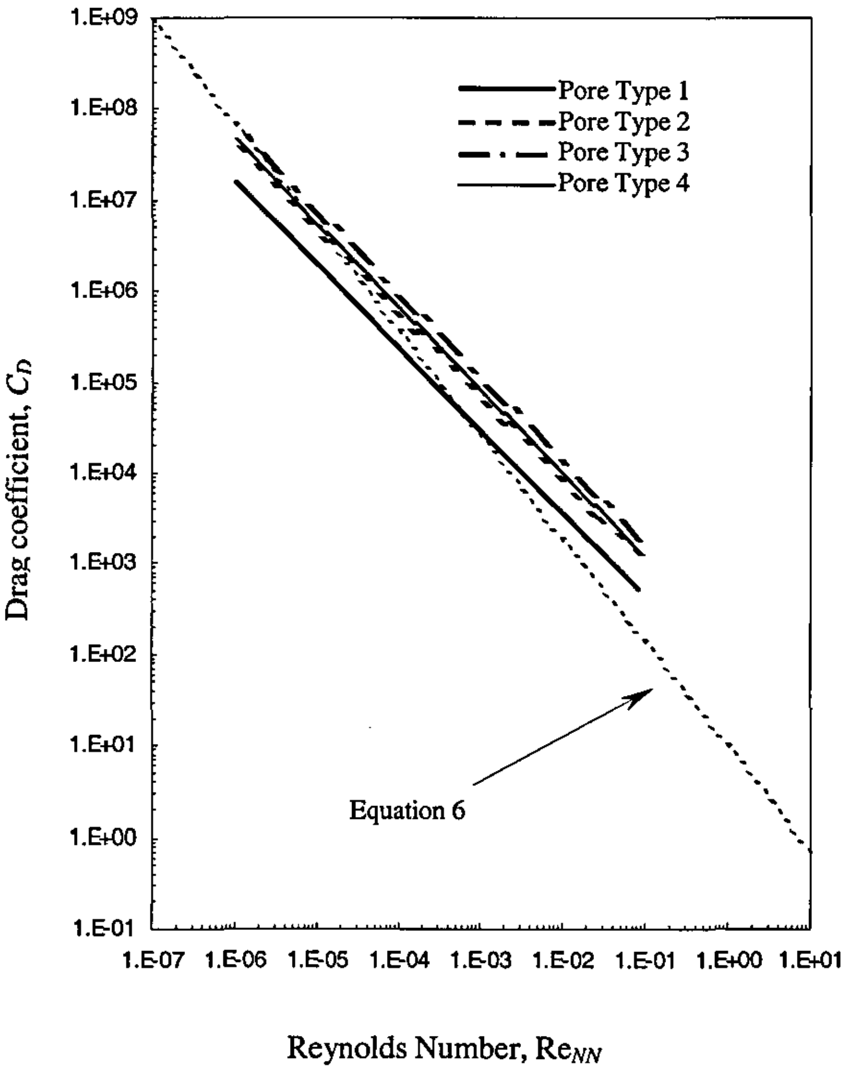


Figure 11 Drag Coefficient *vs* Reynolds Number for non-Newtonian fluid flow through four basic pore types in monofilament cloths and meshes.

CONCLUSION

A 3-D weighted residual finite element scheme for solving Stokes flow has been successfully developed and applied to simulate flow through monofilament wire meshes and fabrics. The accuracy of the model is proven by three test cases of increasing complexity and compared against experimental data for simulation of flow through wire meshes. The pressure drop across the wire mesh was analysed and the results were found to be in good agreement with the existing experimental literature. The results provide a sound basis to proceed with modelling of fluid flow through a wire mesh in order to quantify the effects of pressure drop, stress and particle capture in the domain of filters.

NOMENCLATURE

c_s	Speed of sound in fluid (m s^{-1})
C_D	Drag coefficient
d	Fibre diameter (m)
d_h	Hydraulic diameter (m)
n	Flow behaviour index in Power Law model
p	Pressure (Pa)
Re	Reynolds number
Re_{NN}	Reynolds number for non-Newtonian fluid
t	Time (s)
v_x	Velocity in *x*-direction (m s^{-1})
v_y	Velocity in *y*-direction (m s^{-1})
v_z	Velocity in z-direction (m s^{-1})

Greek Symbols

$\dot{\gamma}$	Strain rate (s^{-1})
β	Porosity
η	Apparent Viscosity (Pa s^n)
η_0	Consistency Index used in Power Law equation
θ	Time stepping parameter
ρ	Density (kg m^{-3})

REFERENCES

Armour, J.C. and Cannon, J.N., 1968. Fluid flow through woven screens. *AIChE journal*, **14**(3): 415-420.

Astarita, G. and Marrucci, G., 1974. Principles of non-Newtonian fluid mechanics. McGraw-Hill, London.

Babuška, I., 1971. Error bounds for finite element method. *Numerical Methods*, **16**:322-333.

Backer, S., 1951. The relationship between the structural geometry of a textile fabric and its physical properties, Part IV: Interstice geometry and air permeability. *Textile Research Journal*, **21**: 703-714.

Bird, R.B., Stewart, W.E. and Lightfoot, E.N., 2002. Transport phenomena. 2nd ed., Wiley, USA.

Brezzi F., 1974. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *RAIRO Analyse Numerique* 8R-2 : 129-151.

Chhabra, R. P. and Richardson, J. F., 1985. Flow of liquids through screens: Relationship between pressure drop and flow rate. *Chemical Engineering Science*, **40**(2): 313-316.

Chhabra, R. P., 1999. Laminar boundary layer heat teamsfer to power-law fluid: An approximate analytical solution. *Journal of Chemical Engineering of Japan*, **32**:812-816.

Chhabra, R. P. and Richardson, J. F., 1999. Non-Newtonian flow in the process industries : fundamentals and engineering applications. Butterworth-Heinemann, Oxford, UK.

Ehrhardt, G., 1983. Flow measurements for wire gauzes. *International Chemical Engineering*, **23**:455-465.

Hanspal, N. S., Ruziwa, W. R., Nassehi, V., Wakeman, R. J., 2004. Finite element modelling of flow of non-Newtonian fluids in pleated cartridge filters. *Proc. of 9th World Filtration Congress*, New Orleans.

Kiljanski, T. and Dziubinski, M., 1996. Resistance to flow of molten polymers through filtration screens. *Chemical Engineering Science*, **51**(19): 4533-4536.

Ladyzhenskaya, O. A., 1969. The mathematical theory of viscous incompressible flow. Gordon and Breach, New York.

Lu, W., Tung, K. and Hwang, K., 1996. Fluid flow through basic weaves of monofilament filter cloth. *Textile Research Journal*, **66**(5): 311-323.

Metzner, A. B., 1956. Non-Newtonian Technology : Fluid Mechanics, mixing, and heat transfer, in *Advances in Chemical Engineering*, Vol 1, edited by Drew T.B. and Hooper J.W. Academic Press, New York.

Nassehi, V., 2002. Practical aspects of finite element modelling of polymer processing. Wiley, Chichester, UK.

Nassehi, V., Hanspal, N. S., Waghode, A. N., Ruziwa, W. R., Wakeman, R. J., 2005. Finite-element modelling of combined free/porous flow regimes: simulation of flow through pleated cartridge filters. *Chemical Engineering Science*, **60**(4): 995-1006.

Rushton, A., 1969. Filtration Research Report. Chem. Eng. Dept., UMIST.

Rushton, A. and Griffiths, P., 1971. Fluid flow in monofilament filter media. *Transactions of the Institutional Chemical Engineers*, **49**: 49-59.

Ruziwa W.R., Hanspal N.S., Waghode A.N., Nassehi V. and Wakeman R.J., 2004. Computer modelling of pleated cartridge filters for viscous fluids. *Filtration*, 4(2): 136-144.

Schowalter, W. R., 1978. Mechanics of non-Newtonian fluids. Pergamon, Oxford, UK.

Skelland, A. H. P., 1967. Non-Newtonian flow and heat transfer. Wiley, New York, USA.

Squiers, J.C., 1984. Fluid flow resistance models for wire weaves. *Filtration and Separation*, **10**:328-330.

Tung, K., Shiau, J., Chuang, C., Li, Y., Lu, W., 2002. CFD analysis on fluid flow through multifilament woven filter cloths. *Separation Science and Technology*, **37**(4): 799-821.

Weighardt, K.E.G., 1953. On the resistance of screens. *Aeronautical Quarterly*, **4**: 186-192.

Zienkiewicz, O. C. and Taylor, R. L., 2000. The finite element method. *Vol 1*. Butterworth-Heinemann, Oxford.

(2) 21ST POLYMER PROCESSING SOCIETY CONFERENCE

June 19 - 23, 2005, Leipzig, Germany

Title: Three-dimensional Finite Element Modelling Of The Convection Dominated Flow of Non-Newtonian Fluid Through a Wire Mesh

Status: Oral presentation.
Presented and published in conference proceedings

Citation:

K. C. Ting, R. J. Wakeman and V. Nassehi, 2005. Three-dimensional Finite Element Modelling Of The Convection Dominated Flow of Non-Newtonian Fluid Through a Wire Mesh, *21st Polymer Processing Society Conference*, Leipzig, Germany.

Three-dimensional Finite Element Modelling Of The Convection Dominated Flow of Non-Newtonian Fluid Through a Wire Mesh

K. C .Ting, V. Nassehi and R. J. Wakeman

Advanced Separation Techniques Group
Department of Chemical Engineering, Loughborough University
Loughborough LE11 3TU United Kingdom

Abstract

A three-dimensional finite element computer model has been developed to simulate non-Newtonian fluid flow through a wire mesh. The governing equations of continuity and momentum were solved by a mixed finite element method and in conjunction with first order Taylor Galerkin scheme for temporal discretization. A slightly perturbed form of continuity equation is considered in this study in order to satisfy the Ladyzhenskaya-Babuska-Brezzi stability condition. The use of this continuity equation, which corresponds to slightly compressible fluids, allows the utilization of equal order interpolation model for the velocity and pressure. The flow of a highly viscous shear-thickening fluid used in aeronautical filters through a wire mesh has been studied. The influence of weave pattern on downstream flow distribution and pressure drop has been investigated and presented in this paper. A commonly used ratio of wire diameter to aperture was adapted and results have been obtained for shear-thickening fluid with power law index $n = 1.3$. The simulation results showed the developed model is capable of generating accurate results in solving three-dimensional non-Newtonian flow problems.

1. Introduction

Woven wire screens have been widely used in the construction of conventional sieves, filters and separators in the solid-liquid processes in the filtration of polymers, chemicals, pharmaceutical, cosmetic, hydraulic oil, fuels, as well as luxury food and beverage industry. Woven wire mesh is used either as the support for finer filter media, for example in filter cartridges, or as the filtering medium itself.

Studies on woven wire screen have been limited and they typically considered permeability, porosity and resistance properties of the wire mesh. Fluid flow through screens have been studied by various researchers such as Wieghardt [1], Armour and Connon [2], Rushton [3], Rushton and Griffiths [4], Ehrhardt [5], Squiers [6], Chhabra and Richardson [7], Kiljanski and Dziubinski [8]. Most of these researchers used various gases and Newtonian liquids as the experimental fluids in their work. For instance, Rushton [3] presented a series of comprehensive experimental data on flow of air and water past filter cloths in a research report, while Rushton and Griffiths [4] presented various approach to explaining the flow situation in woven fabric – those approaches included an orifice analogy, a randomly packed bed analogy and the analytical solution for creeping flow over cylinders.

Lu *et al.* [9] lamented that most previous studies on fluid flow through woven structures have focused on the pressure drop problem and not the macroscopic details of velocity profile and pressure contour in the interstices. They argued that a better understanding of the flow pattern in the woven structure could prove useful in examining the initial stages of cake filtration as well as the effect of weaves on fouling phenomena within a filter cloth. They studied the influence of fabric pore construction on the flow pattern in the interstices and downstream using a CFD software, and discovered that construction of the fabric pores has a significant influence on the flow pattern in the interstices and downstream.

Investigations of flow of non-Newtonian liquids through single screens are very limited. Chhabra and Richardson [7] studied experimentally flow past a single screen for Newtonian liquid and shear-thinning liquids carboxymethyl cellulose (CMC) of various concentration with *n* between 0.34 to 0.61. The correlation between the drag coefficient and the Reynolds number (Re_{NN}) was investigated and reported. Kiljanski and Dziubinski [8] furthered the study to include sets of filtration screens using shear-thinning molten polyethylene for a lower range of *Re* and provided correlation of drag coefficient and *Re* for multiple screens. None of the researchers studied the velocity profile when the fluids flow past the wire mesh interstices.

Ting *et al.* [10] successfully conducted simulations on flow of shear-thinning fluid through a plain weave wire mesh based on the fluid properties reported in Chhabra and Richardson's experiment [7]. Their simulation results showed a close agreement with the experimental works of Chhabra and Richardson for shear-thinning fluid and Rushton [3] for Newtonian fluid.

Shear-thickening fluid forms an important class of Non-Newtonian fluids and today there are an increasing number of engineering fluids characterized by shear-thickening behaviour. One such example is the non-combustible and non-hydrocarbon based aeronautical hydraulic fluid. Considering the significance of the knowledge in the flow behaviour of shear-thickening fluids, it is only natural to take a further step into studying the complex flow field generated by the flow of a shear-thickening fluid through a complex geometry.

Computational fluid dynamics (CFD) techniques provide a powerful and convenient route for the qualitative and quantitative analysis of the non-Newtonian flow systems. With increasingly powerful processors and more economical computing cost, accurate computations of very complex flow problems have been made possible in two-dimensional and even three-dimensional domains. Therefore, CFD seems to be a feasible alternative to study flow past wire mesh as it is normally characterized by small interstices in the range of mm and μm.

There are several numerical analysis techniques commonly employed by engineers to solve the non-linear governing partial differential equations (PDEs). Among these techniques are the Finite Difference Method, Finite Volume Method, Finite Element Method and the Boundary Element Method. In solving non-linear field problems of geometrically complex domain under various types of boundary conditions, Finite Element Method was found to be the most appropriate technique for its flexibility and capability.

2. Mathematical model

A three-dimensional mathematical model based on flow and constitutive equations has been considered. The majority of highly viscous non-Newtonian fluid flows are characterized as low Reynolds number Stokes flow regimes in which the inertia terms in the equation of motion are neglected. In addition, highly viscous flow systems are in general dominated by stress and pressure variations and in comparison the body forces acting upon them are small and can be justifiably ignored. The governing Stokes flow equations in Cartesian coordinate systems are presented as follows:

2.1 Equation of continuity

A slightly perturbed form of continuity equation is considered here and the equation is written as

$$\frac{1}{\rho c_s^{\,2}}\frac{\partial p}{\partial t}+\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \tag{1}$$

where c_s, p, v, ρ represent speed of sound in the fluid, pressure, velocity and density, respectively. This slightly perturbed form of continuity equation has been used in order to satisfy the Ladyzhenskaya-Babuska-Brezzi stability condition [11-13]. The use of this continuity equation, which corresponds to slightly compressible fluids, allows the utilization of equal order interpolation model for the velocity and pressure hence increases the flexibility of the developed solution scheme [14], and has been applied successfully to modelling flows in cartridge filters [15].

2.2 Momentum equation

The majority of highly viscous non-Newtonian fluid flows are characterized as low Reynolds number Stokes flow regimes or creeping flow where the fluid flow is very slow, normally with *Re*<1 [16]. In addition, the body forces acting upon them are small and can be justifiably ignored [14]. The momentum equation can therefore be written as

$$\rho\frac{\partial v_x}{\partial t}=-\frac{\partial p}{\partial x}+\frac{\partial}{\partial x}\left[2\eta\frac{\partial v_x}{\partial x}\right]+\frac{\partial}{\partial y}\left[\eta\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right)\right]+\frac{\partial}{\partial z}\left[\eta\left(\frac{\partial v_z}{\partial x}+\frac{\partial v_x}{\partial z}\right)\right] \quad (2a)$$

$$\rho\frac{\partial v_y}{\partial t}=-\frac{\partial p}{\partial y}+\frac{\partial}{\partial x}\left[\eta\left(\frac{\partial v_x}{\partial y}+\frac{\partial v_y}{\partial x}\right)\right]+\frac{\partial}{\partial y}\left[2\eta\frac{\partial v_y}{\partial y}\right]+\frac{\partial}{\partial z}\left[\eta\left(\frac{\partial v_y}{\partial z}+\frac{\partial v_z}{\partial y}\right)\right] \quad (2b)$$

$$\rho\frac{\partial v_z}{\partial t}=-\frac{\partial p}{\partial z}+\frac{\partial}{\partial x}\left[\eta\left(\frac{\partial v_z}{\partial x}+\frac{\partial v_x}{\partial z}\right)\right]+\frac{\partial}{\partial y}\left[\eta\left(\frac{\partial v_y}{\partial z}+\frac{\partial v_z}{\partial y}\right)\right]+\frac{\partial}{\partial z}\left[2\eta\frac{\partial v_z}{\partial z}\right] \quad (2c)$$

where η is fluid viscosity.

2.3 Constitutive equation

The constitutive equation shows the relationship between the extra stress and the rate of deformation of the fluid. In this study, the viscosity of the non-Newtonian fluid is updated in the solution algorithm using the power law model:

$$\eta=\eta_0(\dot{\gamma})^{n-1} \quad (3)$$

where η is the apparent viscosity, η_0 is the consistency index, n is the power law index and $\dot{\gamma}$ is the strain rate.

2.4 Numerical scheme

The numerical scheme used in this work is the Mixed Finite Element U-V-W-P technique [17] in which both velocity and pressure in the governing equations are regarded as primary variables and are discretized as unknowns. The first order Taylor-Galerkin method was used to discretize the equations.

3. Model validation

3.1 Computational mesh

Three-dimensional test cases were devised to check the computer codes developed. Three test cases were designed in increasing mesh complexity to progressively test the capability of the algorithm. *Test Case 1* was used to validate the capability of the program in solving the continuity and motion equations. The main test for the accuracy of the simulation results is ensuring a mass balance over the domain under investigation. A rectangular domain, shown in *Figure 1*, was adapted with four non-permeable no-slip vertical walls and the inlet from upper horizontal surface and outlet at bottom horizontal surface. The ratio of domain's length (x), width (y) and height (z) is 3:3:5.

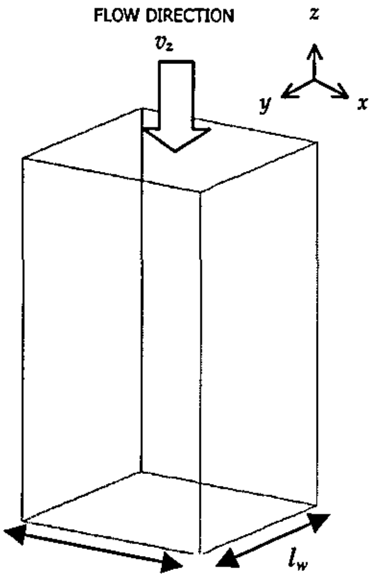


Figure 1: A schematic diagram of *Test Case 1.*

Test case 2 is slightly more complex with two half-cylinders protruding in a rectangular domain, emulating the flow around wire meshes. The ratio of aperture size to wire diameter used in this test case is 1.5 (values from about 1.1 up to about 7 are commonly used ratios in industrial wire meshes). The inlet is again from upper horizontal surface and outlet at bottom horizontal surface. The half cylinder surfaces were considered to be non-permeable and non-slip. The ratio of domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:2:2 and the schematic computational mesh is shown in *Figure* 2.

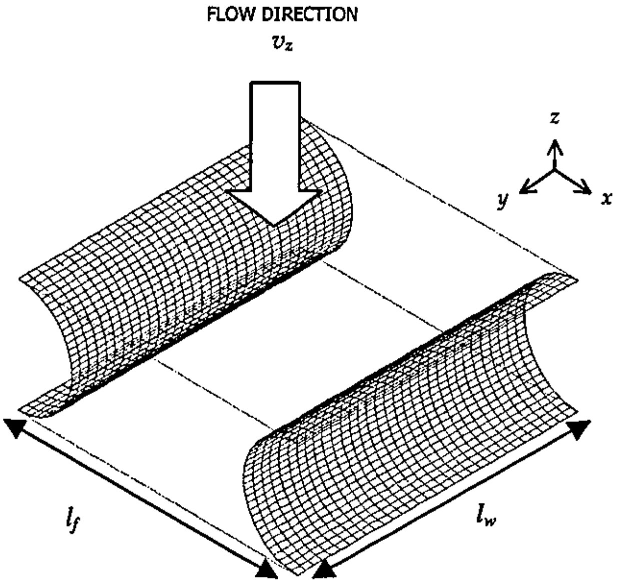


Figure 2: A schematic diagram of *Test Case* 2.

Test Case 3 has all the basic features of a typical wire mesh domain except that the four supposedly intertwined weft and warp wires were flattened to a horizontal plane to test the response of the model to highly complex geometry and mesh distortion. The four cylinders were jointed together via a 45° slice at each corner. The ratio of domain's length (x), width (y), height (z) and wire diameter (d) is 5:5:2:2. The results from this test case provide a powerful pretext into understanding the complex flow field in different pore types of a wire mesh. Inlet and outlet settings are similar to *Test Case 1* and *Test Case 2*. The half cylinder surfaces were again considered to be non-permeable and non-slip. The schematic domain is shown in *Figure* 3.

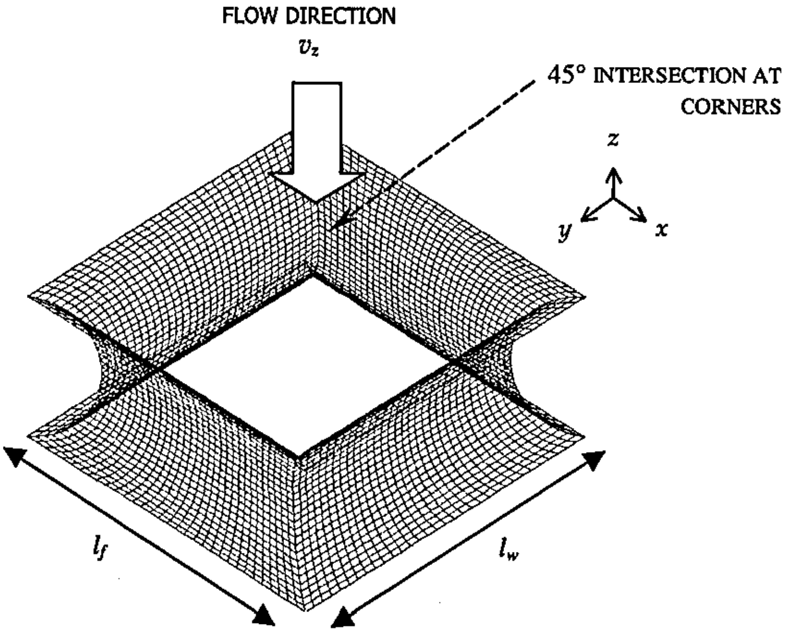


Figure 3: A schematic diagram of *Test Case 3.*

3.2 Boundary conditions

The velocity v_z corresponds to the approach velocity of the fluid towards the aperture and was set as 0.1ms^{-1} at the inlet. The boundaries (the wire meshes filaments) were considered to be impermeable, non-slip surfaces where $v_x = v_y = v_z = 0$. The computational mesh and degree of freedom (DOF) used for each test case are given in *Table 1.*

Table 1: Computational mesh used for each test cases.

Test Case	No. of Element	No. of Nodes	DOF
1	1920	2511	3544
2	4096	4913	4572
3	4096	4913	4164

3.3 Physical properties and numerical parameter

A shear-thickening non-Newtonian fluid with the property given in *Table 2* was used in the simulations. Similar properties were used for the simulations of the Newtonian fluid, but with n taken to be 1.0.

Table 2: Physical properties of the shear-thickening fluid used.

Physical Property	value
Density, ρ	970 kg m^{-3}
Consistency index, K	80.0 kg m^{-1}s^{-1}
Power Law index, n	1.3

A time step of Δt = 0.01s was used throughout the simulations and the θ used for the flow modelling is 0.95. An Intel Pentium IV 2.6 GHz processor has been used to execute the computation to produce simulation results. The pre-processing work of the domain development and the mesh generation were handled using COSMOS™ GEOSTAR. Post-processing presentation of field variables such as flow velocity vectors and pressure contour were plotted using commercial graphic software package SURFER® 8.

4. Validation results

Figure 4 illustrate the velocity vectors of *Test Case 1* in the *x-z* plane at $y = l_w/2$ for non-Newtonian fluid. The outlet velocity of non-Newtonian fluid is observed to be about 0.3% lower than that of the Newtonian liquid, largely due to the shear-thickening effect of the non-Newtonian fluid. The overall mass balance for Newtonian and non-Newtonian simulations is 99.7% and 99.9%, respectively. The excellent accuracy showed that continuity was preserved in the simulations.

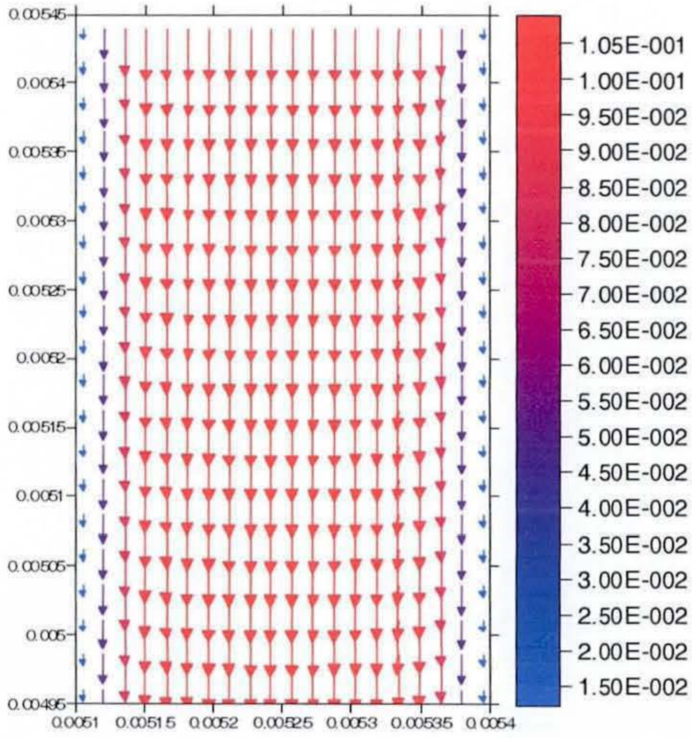


Figure 4: The velocity vectors of test case 1 in the *x-z* plane at $y = l_w/2$ for non-Newtonian fluid.

Figure 5 shows a typical velocity vector of *Test Case 2* for Newtonian and non-Newtonian fluid. Higher velocities were observed at the area immediately above the cylinders and at the section between the cylinders compared to the velocity at the inlet and outlet. This is attributed to the compression of the fluid that leads to the acceleration of the fluid velocity. Again, the shear-thickening effect of the non-Newtonian fluid retarded the fluid flow by about 0.3%. The overall mass balance for Newtonian and non-Newtonian simulations in *Test Case 2* showed an outlet flow of 95.0% and 95.2% compared to the inlet flow. The discrepancy is due to the relatively coarse mesh used and can be improved by further mesh refinement. This is evident when an earlier run with a very coarse mesh of 512 elements gave a mass balance of 86.2% for Newtonian liquid.

Figure 6 depicted a typical simulation result of *Test Case 3* for Newtonian and non-Newtonian fluid. The highly complex geometry of the four adjoined cylinders formed a narrow channel for the fluid to flow through. The maximum *z*-direction velocity v_z of non-Newtonian fluid were found to be 4.7% lower compared to that of Newtonian fluid. This observation is consistent with the results obtained for *Test Case 1* and *Test Case 2*. The mass balance for Newtonian and non-Newtonian simulations is 96.1% and 96.4%, respectively. Simulations using a very coarse mesh of 512 elements gave a mass balance of 86.3%. These results shows that the mass balance accuracy can be improved by mesh refinement. The overall consistency in the results showed that the model is capable of solving three-dimensional flow problem in a domain with complex geometry.

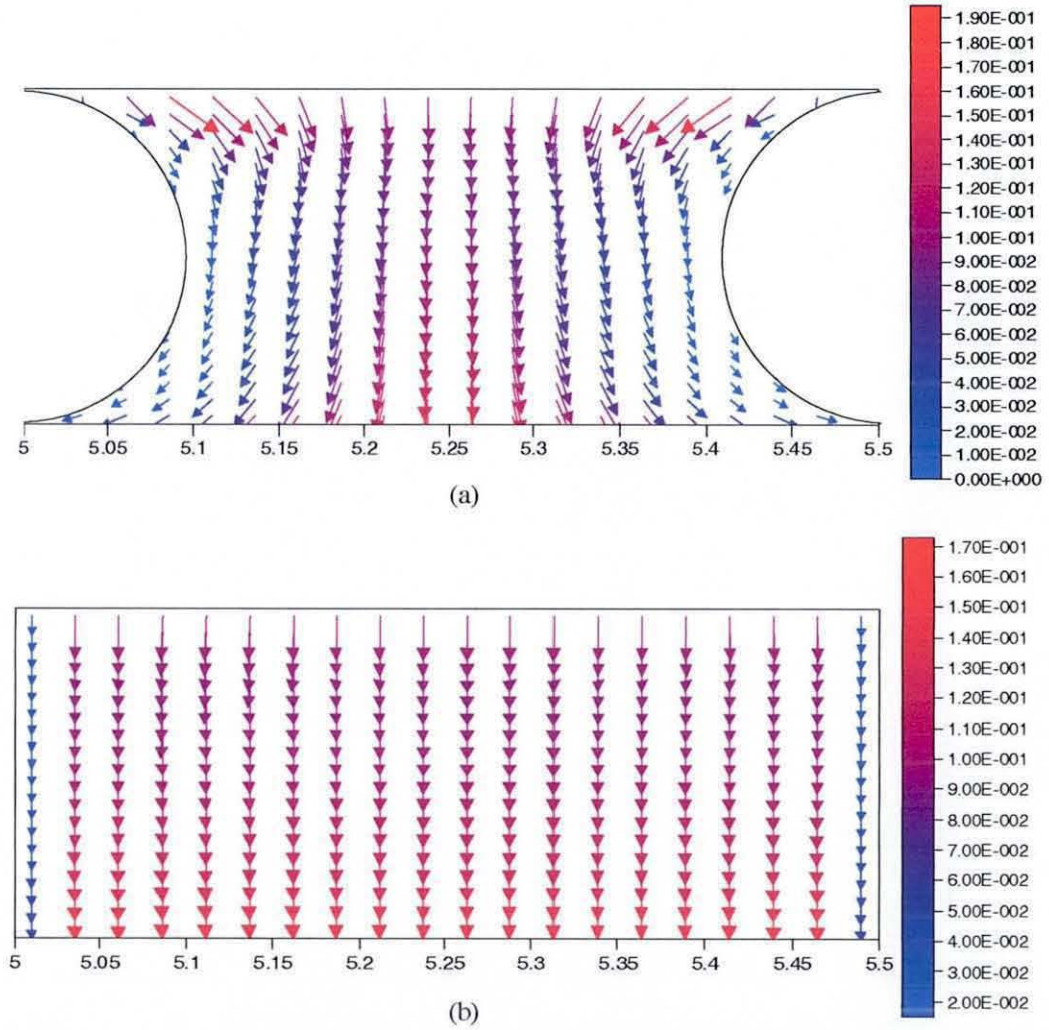
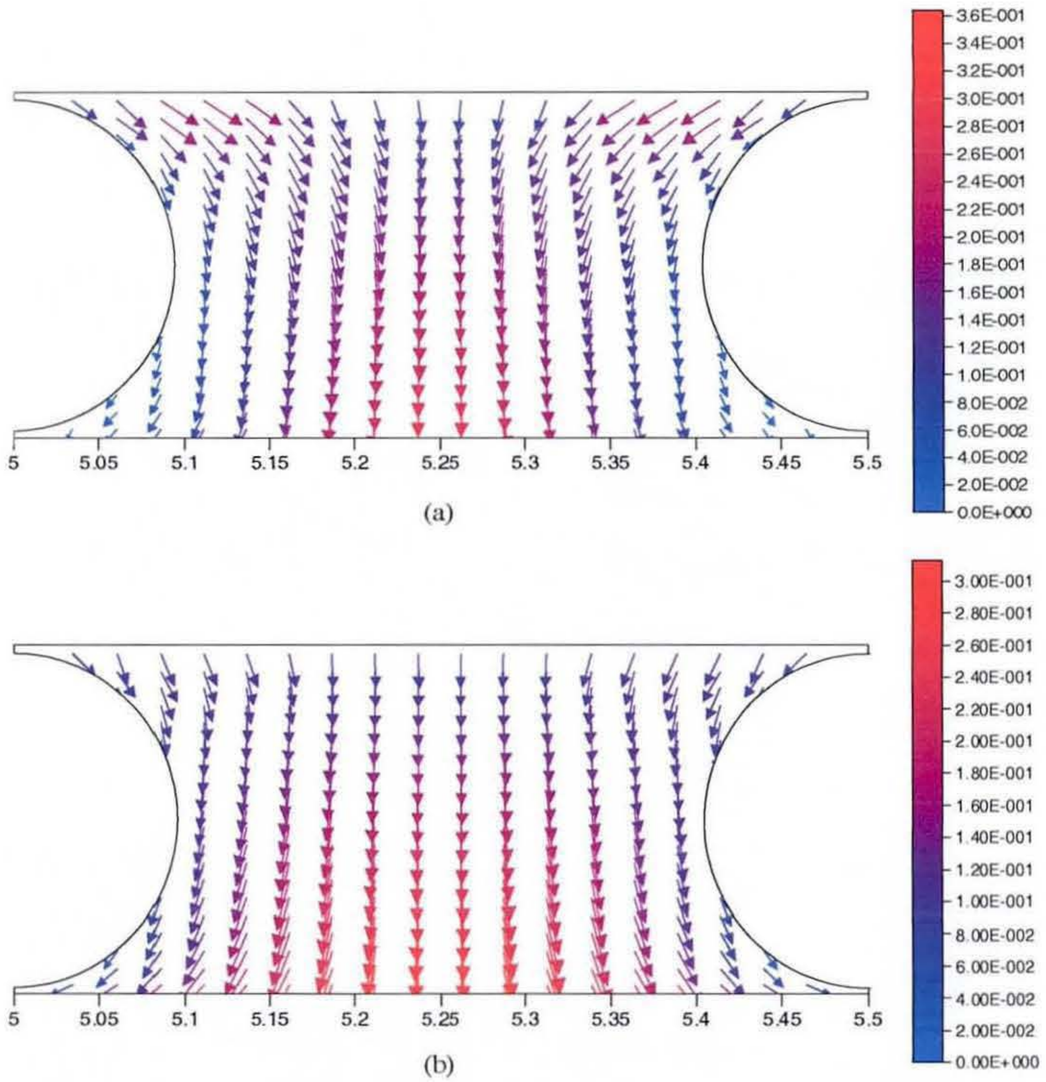


Figure 5: The velocity vectors of test case 2 in the (a) *x*-*z* plane at $y = l_w/2$
(b) *y*-*z* plane at $x = l_l/2$ for non-Newtonian fluid.



(a)

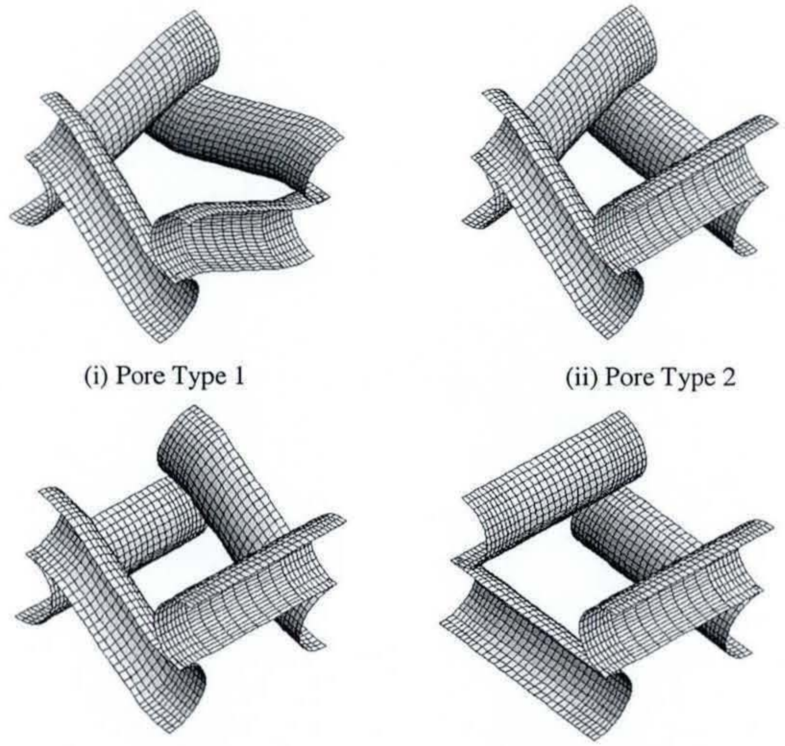
(b)

Figure 6: The velocity vectors of test case 3 in the (a) *x*-*z* plane at $y = l_w/2$ (b) *y*-*z* plane at $x = l_f/2$ for non-Newtonian fluid.

5. Results and discussions

The model can now be applied to the simulation of flow through a monofilament woven wire mesh. The weaves considered in this work are the plain weave, the twill weave, and the satin weave. These give rise to the four types of basic pore shape, as suggested by Backer [18] and shown in *Figure 8*, where only the half circle of each monofilament wire is illustrated.

Fluid flow past the four basic pore types illustrated in *Figure 7* was investigated. A schematic representation of the simulation domain for pore type 1 is given in *Figure 8*. The ratio of aperture size to wire diameter used is 1.5 and the ratio of domain's length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:5:2. The flow inlet is at the top horizontal surface while the flow exit is at the lower horizontal surface. An open boundary is taken for the four sidewalls where no boundary condition was specified. The monofilament cut-sized wires were assumed to be non-permeable and non-slip. Computational grids with properties given in *Table 3* were used in the simulations.



(i) Pore Type 1

(ii) Pore Type 2

(iii) Pore Type 3

(iv) Pore Type 4

Figure 7: Four types of basic weave patterns of a wire mesh.

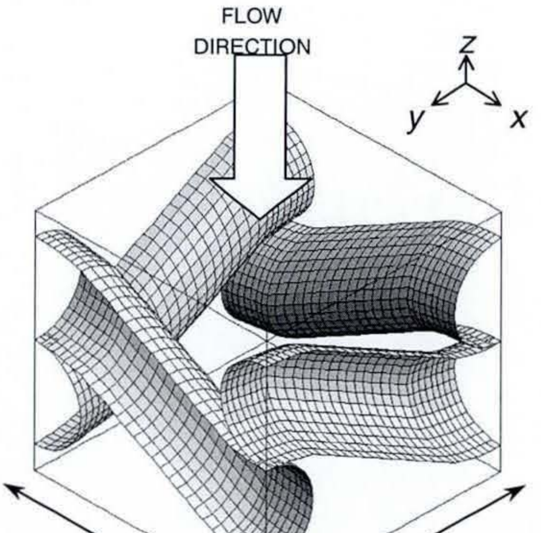
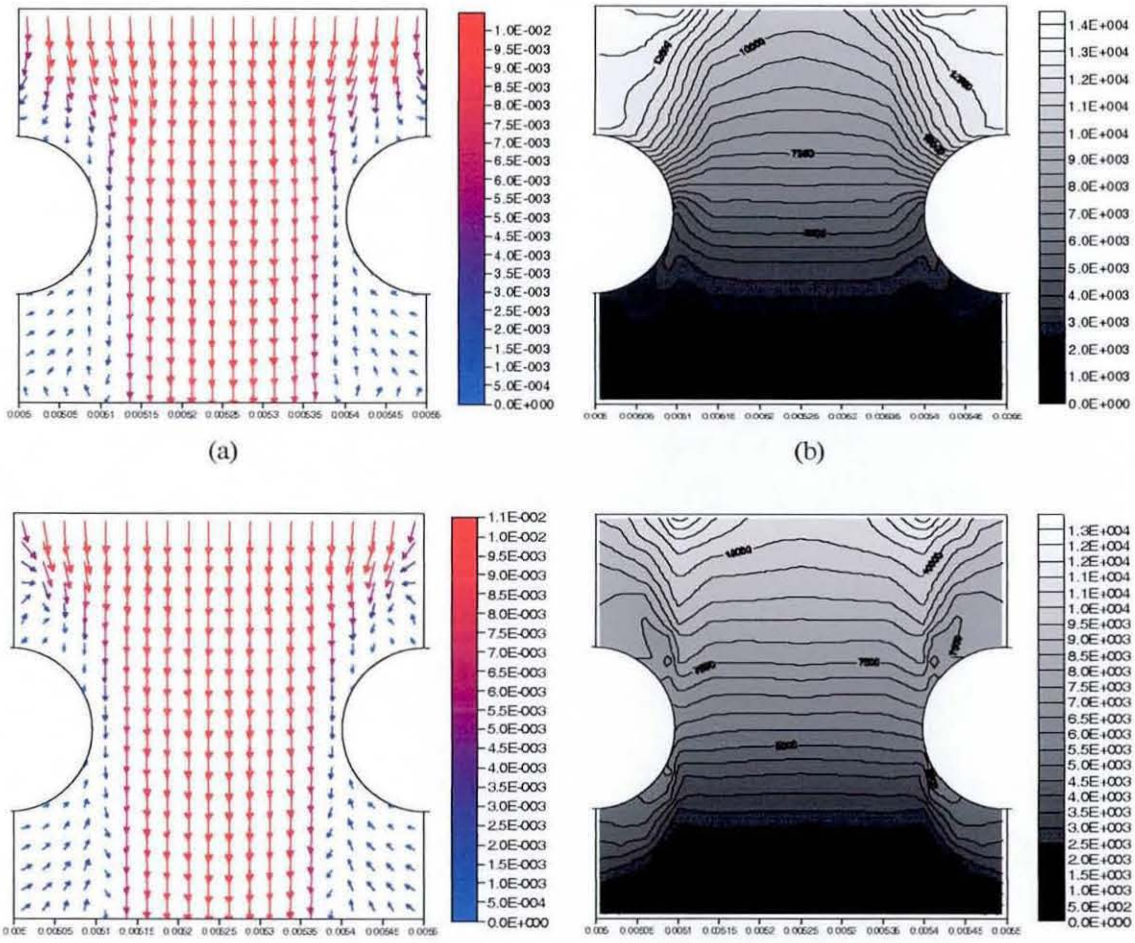


Figure 8: Schematic diagram of simulation domain for basic pore type 1.

Table 3: Properties of the finite element domains used in the simulations of four basic pore types.

Pore Type	No. of Element	No. of Nodes	DOF
1	4608	6560	3363
2	4608	6514	3288
3	4608	6527	3282
4	4608	6437	3210



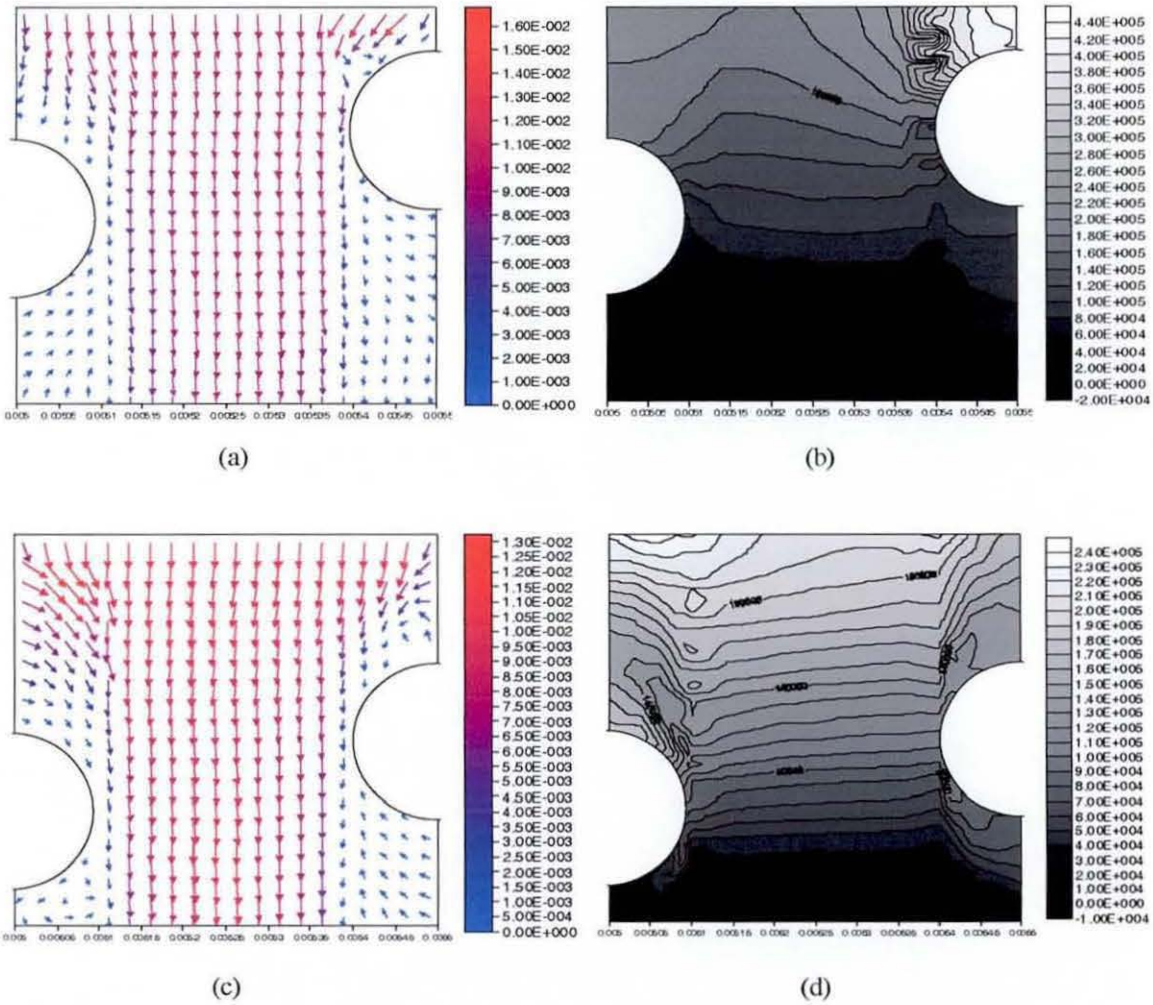
(a)

(b)

(c)

(d)

Figure 9: Flow of shear-thickening fluid in the interstices of pore type 1 showing (a) velocity field (b) pressure contour in x-z plane at $y = l_w/2$; and (c) velocity field (d) pressure contour in y-z plane at $x = l_f/2$.



(a) (b)

(c) (d)

Figure 10: Flow of shear-thickening fluid in the interstices of pore type 2 showing (a) velocity field (b) pressure contour in x-z plane at $y = l_w/2$; and (c) velocity field (d) pressure contour in y-z plane at $x = l_f/2$.

Figure 9, 10, 11 and 12 showed the velocity vectors and the pressure contours of pore type 1, 2, 3 and 4 in the x-z plane at $y = l_w/2$ and in y-z plane at $x = l_f/2$, respectively. The cross sections of pore type 1 is symmetrical at $y = l_w/2$ and $x = l_f/2$. This is reflected in the velocity distributions and the pressure contour at these points shown in *Figure 9*. Pore type 2 has the most unsymmetrical geometry among the four basic pore types. This is evident in the velocity and pressure plot illustrated in *Figure 10* where an absence of symmetrical pattern in both planes is observed. Pore type 3 has a symmetrical geometry at $x = l_f/2$ but not at $y = l_w/2$. This is reflected in *Figure 11*. The cross sections of pore type 4 are symmetrical in both at $y = l_w/2$ and $x = l_f/2$ as in the case of pore type 1. The velocity distribution and the pressure contours showed in *Figure 12* are in perfect symmetry consistent with the symmetrical geometry.

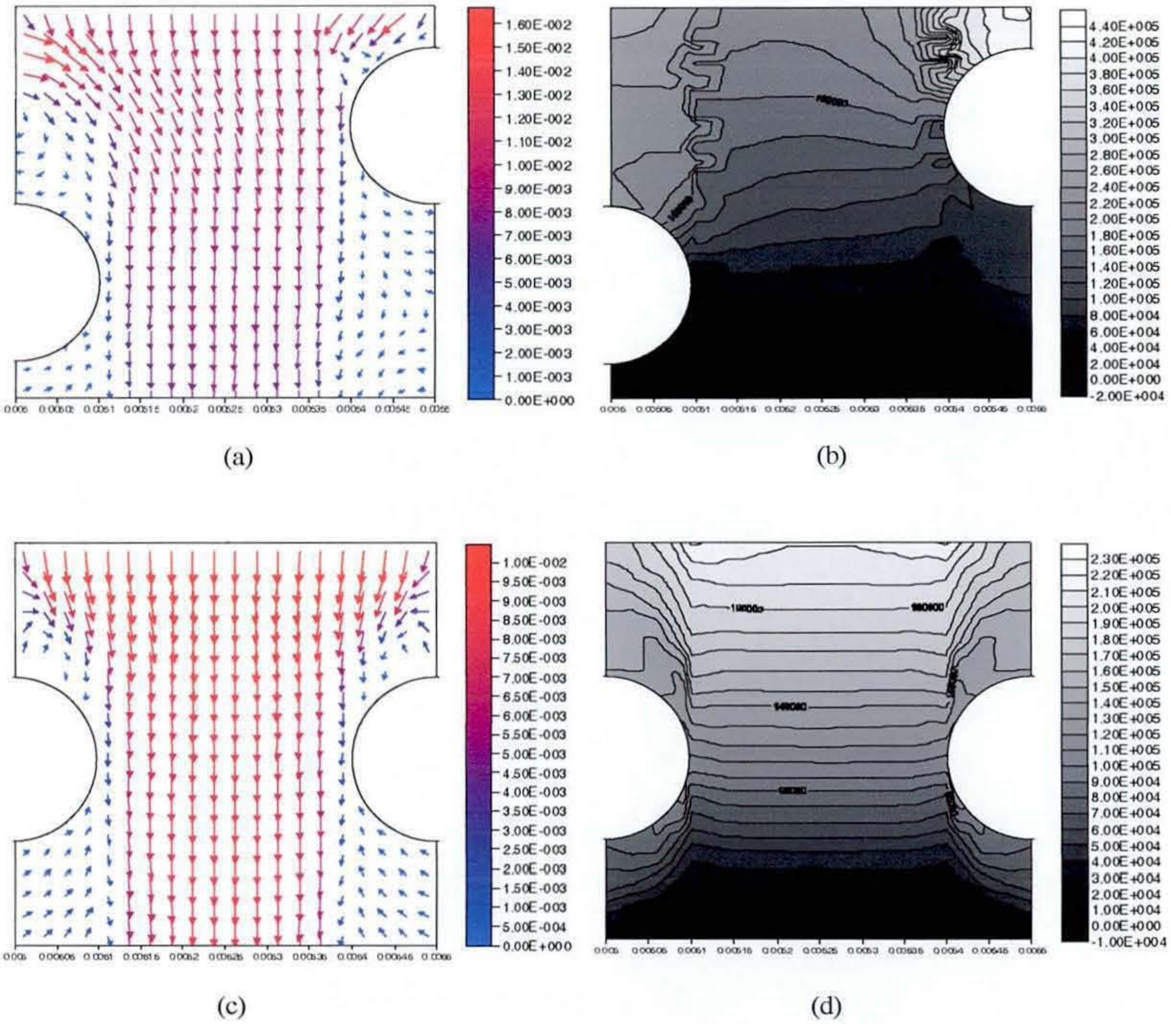
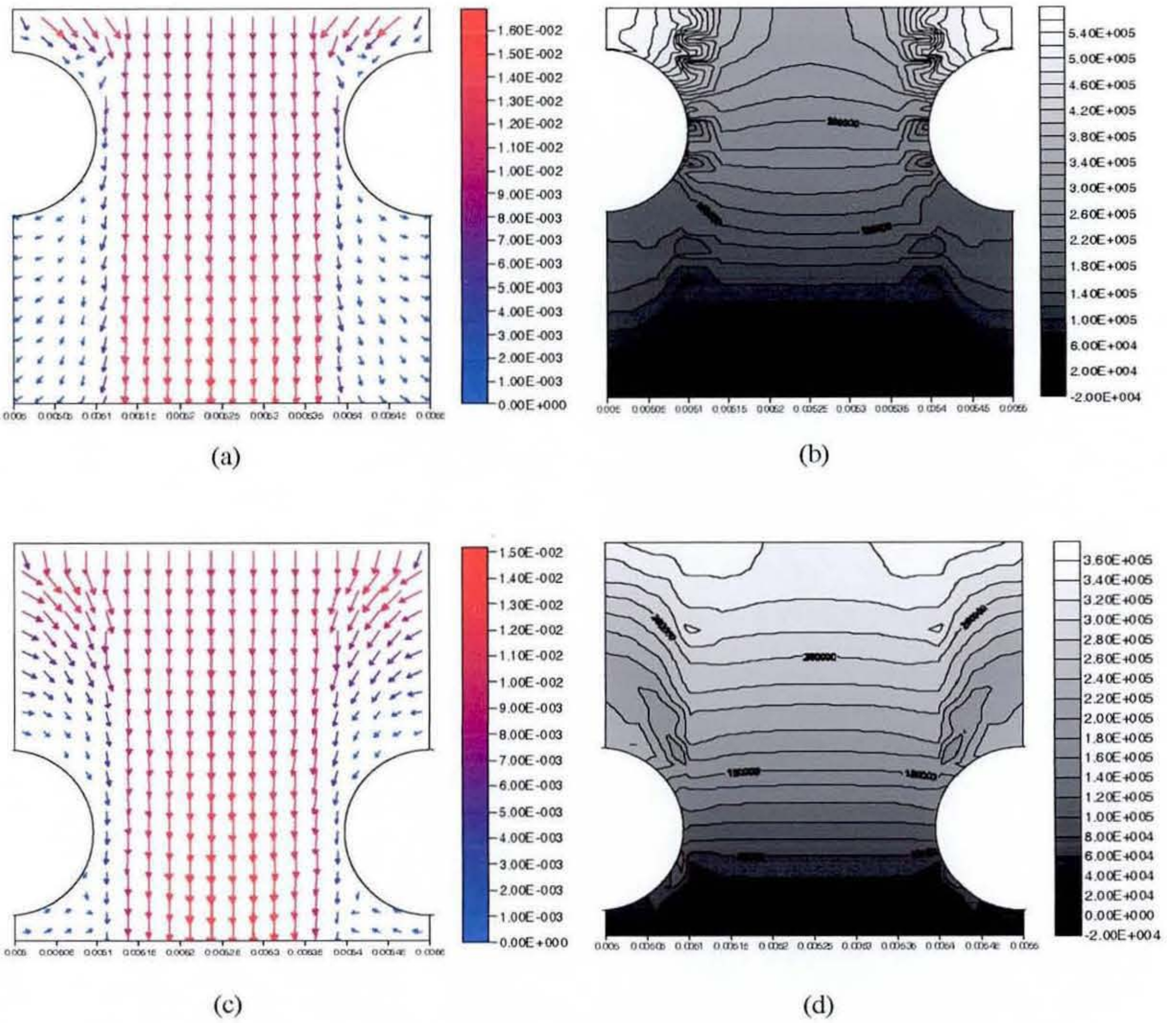


Figure 11: Flow of shear-thickening fluid in the interstices of pore type 3 showing (a) velocity field (b) pressure contour in x-z plane at $y = l_w/2$; and (c) velocity field (d) pressure contour in y-z plane at $x = l_f/2$.

The velocity distributions in *Figure 9, 10, 11 and 12* showed a higher velocity at the area immediately above the wire mesh and at the section between the wire mesh where highly complex geometry of the four monofilament wires formed a narrow channel for the fluid to flow through. In each case, the flow is observed to swirl around the upper surfaces of the upper wire meshes into the narrow channel in each pore type on both x-z and y-z planes. This is consistent with the location of the wire mesh on both planes. The upper wires in each pore type also noticeably affected the flow pattern in the interstices and the velocity distribution downstream. This observation is important in the selection of weave patterns for various filtration applications as the lower lying wire surface will be noticeably fouled first at the initial stages of the filter cake formation.

The pressure contours in *Figure 9, 10, 11 and 12* showed a higher pressure being asserted on the upper surface of the wires by the oncoming fluid. This is conceivable as the upper surfaces of the wire meshes are directly normal to the direction of the flow therefore offering more resistance to the fluid flow.



(a) (b)

(c) (d)

Figure 12: Flow of shear-thickening fluid in the interstices of pore type 4 showing (a) velocity field (b) pressure contour in x-z plane at $y = l_w/2$; and (c) velocity field (d) pressure contour in y-z plane at $x = l_f/2$.

The pressure in all the pore types were observed to drop rapidly as the fluid flows through the interstices of the wire meshes. This is similar in essence to the pressure drop across a filter media. Interesting, the pressure drop for each pore type is observed to be different. Pore type 1 was observed to give the lowest pressure drop, followed by pore type 2 and 3. Pore type 4 was observed to give the highest pressure drop compared to other basic pore types.

Chhabra and Richardson [7] presented their experimental results for shear-thinning fluid using a loss coefficient C_D (which we refer to as a drag coefficient) defined as

$$C_D = \frac{2\Delta p}{\rho (u/\varepsilon)^2} \cdot \frac{1}{(1-\varepsilon)} \tag{4}$$

where ε is the porosity of the wire mesh and Δp the pressure drop across the wire mesh. They also used a modified Reynolds number for non-Newtonian fluid Re_{NN} ($= \left|\rho (u/\varepsilon)^{2-n} d^n \right| / \eta_0$), a form similar to that used by Metzner [19] and Skelland [20] with porosity taken into consideration. Ting *et al.*[10] pointed out that Chhabra and Richardson had erroneously taken d to be wire diameter rather than the hydraulic diameter, d_h, which should be taken as the aperture in this case. A more appropriate correlation for Re_{NN} consistent to that used by earlier researchers has been given as [10]

$$Re_{NN} = \frac{\rho (u/\varepsilon)^{2-n} d_h^{\ n}}{\eta_0} \tag{5}$$

where Re_{NN} reduces to the Reynolds number for a Newtonian liquid as n becomes unity. Based on Eqs. (4) and (5), Ting *et al.*[10] recalculated the experimental results presented in Chhabra and Richardson's paper, as presented in *Figure 13*. They also analysed the experimental data reported by Rushton [3] on water flow through monofilament Nylon cloth using Eqs. (4) and (5) and included in *Figure 13*. For simulations in this work, the relationship between the flow rate and the pressure drop for each fluid is calculated from the model and presented using Eqs. (4) and (5) and plotted in *Figure 13*. The drag coefficient C_D and the Reynolds number Re_{NN} seems to be very close to the best fit curve reported by Ting *et al.*[10] as

$$C_D = \frac{10}{{Re_{NN}}^{1.14}} \tag{6}$$

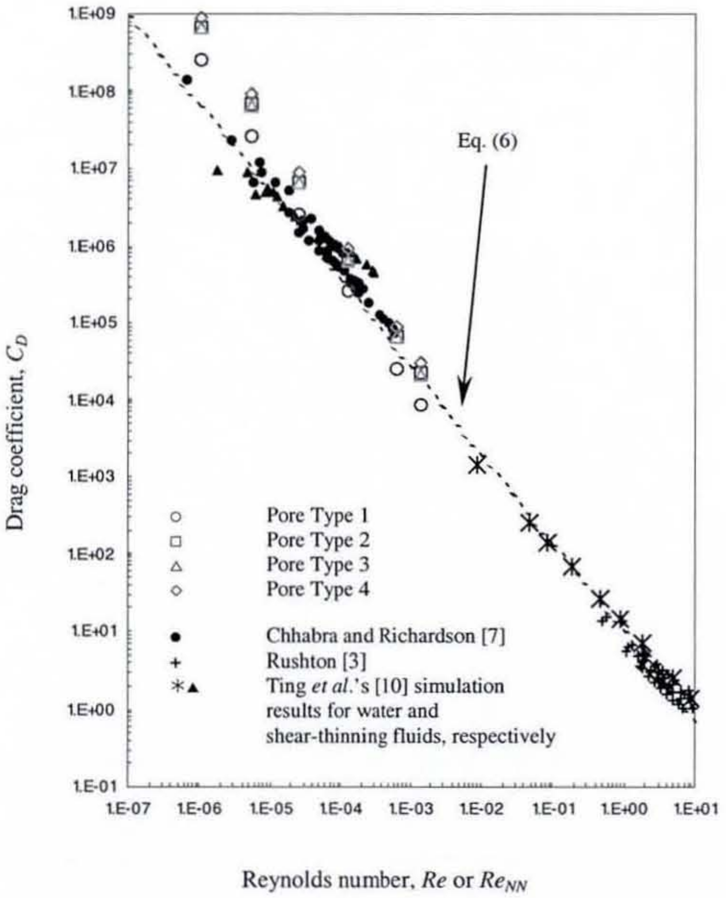


Figure 13: Drag-coefficient –Reynolds number relationship for shear-thickening liquids.

6. Conclusion

A three-dimensional weighted residual finite element scheme for solving Stokes flow has been successfully developed and applied to simulate flow of a highly viscous shear-thickening fluid through four basic pore types in a wire mesh. The accuracy of the model has been validated by three test cases of increasing geometrical complexity. The influence of weave pattern on downstream flow distribution and pressure drop has been successfully studied and compared using cross sections of velocity distributions and pressure contours for each basic pore type. Pressure drops across each pore type were analysed and pore type 1 was observed to give the lowest pressure drop while pore type 4 gives the highest pressure drop. The simulation results in this study showed that the developed model is capable of simulating non-Newtonian fluid flow through a complex three-dimensional domain.

Nomenclature

c_s	Speed of sound in fluid (ms^{-1})
C_D	Drag coefficient
d	Diameter (m)
d_h	Hydraulic diameter (m)
n	Flow behaviour index in Power Law model
p	Pressure (Pa)
Re	Reynolds number
Re_{NN}	Reynolds number for non-Newtonian fluid
t	Time (s)
v_x	Velocity in x-direction (m/s)
v_y	Velocity in y-direction (m/s)
v_z	Velocity in z-direction (m/s)

Greek Symbols

$\dot{\gamma}$	Strain rate (s^{-1})
ε	Porosity
η	Apparent viscosity ($Pa.s^n$)
η_0	Consistency index used in Power Law equation
θ	Time stepping parameter
ρ	Density ($kg.m^{-3}$)

References

1. Wieghardt, K.E.G., *On the Resistance of Screens*, Aero. Quart., 4, pp. 186-192 (1953)
2. Armour, J.C.; Cannon, J.N., *Fluid Flow Through Woven Screens,* AIChE journal, 14, 3, pp. 415-420 (1968)
3. Rushton, A., *Filtration Research Report,* Chem. Eng. Dept., UMIST, (1969)
4. Rushton, A.; Griffiths, P., *Fluid Flow in Monofilament Filter Media,* Trans. Instn. Chem. Engrs., 49, pp. 49-59 (1971)
5. Ehrhardt, G., *Flow Measurements for Wire Gauzes*, Int. Chem. Engng., 23, pp. 455-465 (1983)
6. Squiers, J.C., *Fluid Flow Resistance Models for Wire Weaves*, Filtr. Sep., 10, pp. 328-330 (1984)
7. Chhabra, R. P.; Richardson, J. F., *Flow of Liquids Through Screens: Relationship Between Pressure Drop and Flow Rate*, Chem. Eng. Sci., 40, 2, pp. 313-316 (1985)
8. Kiljanski, T.; Dziubinski, M., *Resistance to Flow of Molten Polymers Through Filtration Screens*, Chem. Eng. Sci., 51, 19, pp. 4533-4536 (1996)
9. Lu, W.; Tung, K.; Hwang, K., *Fluid Flow Through Basic Weaves of Monofilament Filter Cloth*, Text. Res. J., 66, 5, pp. 311-323 (1996)

10. Ting, K.C.; Nassehi, V.; Wakeman R.J., *Modelling Flow in Monofilament Cloths – Prediction of Pressure Loss*, Proc. Of Filt. Soc. Conf.: media for gas and liquid filtration applications (2005)
11. Ladyzhenskaya, O. A., *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach (1969)
12. Babuska, I., *Error Bounds for Finite Element Method*, Numer. Methods, 16, pp. 322-333 (1971)
13. Brezzi, F., *On the Existence, Uniqueness and Approximation of Saddle-Point Problems Arising from Lagrangian Multipliers*, RAIRO, Serie Rouge 8R-2, pp. 129-51 (1974)
14. Nassehi, V., *Practical Aspects of Finite Element Modelling of Polymer Processing*, Wiley (2002)
15. Ruziwa W.R.; Hanspal N.S.; Waghode A.N.; Nassehi V.; Wakeman R.J., *Computer Modelling of Pleated Cartridge Filters for Viscous Fluids*, Filtration, 4, 2, pp. 136-144 (2004)
16. Bird, R.B.; Stewart, W.E.; Lightfoot, E.N., *Transport Phenomena*, 2nd ed., Wiley (2002)
17. Zienkiewicz, O. C.; Taylor, R. L., *The Finite Element Method*, Vol 1, Butterworth-Heinemann (2000)
18. Backer, S., *The Relationship Between the Structural Geometry of a Textile Fabric and Its Physical Properties, Part IV: Interstice Geometry and Air Permeability*, Text. Res. J., 21, pp. 703-714 (1951)
19. Metzner, A. B., *Non-Newtonian Technology : Fluid Mechanics, Mixing and Heat Transfer*, in Advances in Chemical Engineering, Vol 1, edited by Drew T.B. and Hooper J.W, Academic Press (1956)
20. Skelland, A. H. P., *Non-Newtonian Flow and Heat Transfer*, Wiley (1967)

(3) FILTECH 2005

Oct 11 - 13, 2005, Wiesbaden, Germany

Title: Finite element modelling of flow through plain, twill and satin weave monofilament cloths

Status: Accepted for oral presentation.

Citation:

R. J. Wakeman, **K. C. Ting**, and V. Nassehi, 2005. Finite element modelling of flow through plain, twill and satin weave monofilament cloths, *FILTECH 2005 (International Conference & Exhibition for Filtration and Separation Technology)*, Wiesbaden, Germany. (accepted for oral presentation)

FINITE ELEMENT MODELLING OF FLOW THROUGH PLAIN, TWILL AND SATIN WEAVE MONOFILAMENT CLOTHS

R.J. Wakeman, K.C. Ting and V. Nassehi
Advanced Separation Technologies Group,
Department of Chemical Engineering, Loughborough University,
Loughborough, LE11 3TU, UK.

ABSTRACT

Monofilament filter cloths are used as the separation media in filtration; woven wire cloths or screens are also used as the media in filters or to enhance the integrity of the filter medium in, for example, filter cartridges. In this paper we present research results aimed at simulating non-Newtonian fluid flow through a woven cloth. Due to the complex geometry of a woven cloth, 3-D modelling is necessary to visualise correctly the structure of the flow and hence to predict pressure losses. Modelling in a 3-D domain was done using a finite element method which is known to cope with flow domains in complex geometries very effectively. The simulation results were found to be in good agreement with experimental data, showing the developed model to be capable of generating accurate results for flow of both Newtonian and non-Newtonian fluids through filter media.

Keywords: Woven filter media, woven screens, CFD simulation, finite element method, modelling, Newtonian fluids, non-Newtonian fluids, pressure loss.

Woven fabrics and wire cloths are used as filter media, and metal and plastic screens are also widely used in the construction of filters and separators, as either the support for finer filter media or as the filtering medium. Fluid flow through cloths and screens has been studied experimentally (Wieghardt, 1953; Armour and Connon, 1968; Rushton, 1969; Rushton and Griffiths, 1971; Ehrhardt, 1983; Squiers, 1984; Chhabra and Richardson, 1985; Kiljanski and Dziubinski, 1996). Most of these used gases and Newtonian liquids as the experimental fluids.

Investigation of flow of non-Newtonian liquids through single screens are few. Chhabra and Richardson (1985) studied flow through a screen using a Newtonian liquid and shear-thinning carboxymethyl cellulose (CMC) solutions of various concentration with *n* (the Power Law index) between 0.34 and 0.61. Kiljanski and Dziubinski (1996) furthered the study to include filter screens using shear thinning molten polyethylene for a lower range of *Re* and provided a correlation of drag coefficient with *Re* for multiple screens. None of these researchers studied the velocity profile when the fluids flow through the screen.

Lu *et al.* (1996) lamented that most previous studies on fluid flow through woven structures have focused on the pressure drop problem and not the macroscopic details of the velocity profile and pressure contour in the weave apertures. They argued that a better understanding of the flow pattern in the woven structure could prove useful in examining the initial stages of cake filtration as well as the effect of weaves on fouling phenomena within a filter cloth, and studied the influence of fabric pore construction on the flow pattern in the interstices and downstream of a cloth using the CFD software FLUENT, and discovered that the flow pattern in the interstices and downstream were different for each basic pore type. As the position of the upper filament in each pore type differs, the flow of water which swirls around the upper surfaces of the upper filament into the narrow channel between the cloth filaments was characteristically unique for each case. Tung *et al.* (2002) furthered the study to include fluid flow through multifilament cloths and spun staple yarns. Their simulations results showed a similar pattern of pressure drop to that of Lu *et al.* where pore type 1 (see Figure 1) was found to give the highest pressure drop and pore type 4 gives the lowest pressure drop.

MATHEMATICAL MODEL

A 3-D mathematical model based on flow and constitutive equations has been developed. The majority of highly viscous non-Newtonian fluid flows are characterized as low Reynolds number Stokes flow regimes in which the inertia terms in the equation of motion are neglected. In addition, highly viscous flow systems are in general dominated by stress and pressure variations and the body forces acting on the fluid are relatively small and can be justifiably ignored. The governing Stokes flow equations in Cartesian coordinate systems are summarised as:

Equation of Continuity The continuity equation for an incompressible fluid is written as $\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0$. To satisfy the Ladyzhenskaya-Babuska-Brezzi stability condition, the slightly perturbed form of this equation is used here is:

$$\frac{1}{\rho c_s^2}\frac{\partial p}{\partial t}+\frac{\partial v_x}{\partial x}+\frac{\partial v_y}{\partial y}+\frac{\partial v_z}{\partial z}=0 \tag{1}$$

where c_s, p, v, ρ represent the speed of sound in the fluid, pressure, velocity and the fluid density, respectively.

The use of this continuity equation, which corresponds to slightly compressible fluids, allows the utilization of an equal order interpolation model for the velocity and pressure, and has been applied successfully to modelling flows in pleated cartridge filters (Ruziwa *et al.*, 2004; Hanspal *et al.*, 2004; Nassehi *et al.*, 2005).

Momentum Equation The momentum equation can be written as

$$\rho\frac{\partial v_i}{\partial t}=-\frac{\partial p}{\partial i}+\frac{\partial}{\partial i}\left[2\eta\frac{\partial v_i}{\partial i}\right]+\sum\frac{\partial}{\partial i}\left[\eta\left(\frac{\partial v_i}{\partial j}+\frac{\partial v_j}{\partial i}\right)\right] \tag{2}$$

where $i = x, y, z$ and $j \neq i$, and η is fluid viscosity.

Constitutive Equation The viscosity of the non-Newtonian fluid is updated in the solution algorithm using the power law model:

$$\eta=\eta_0\left(\dot{\gamma}\right)^{n-1} \tag{3}$$

where η is the apparent viscosity, η_0 is the consistency index, n is the power law index and $\dot{\gamma}$ is the shear rate.

Numerical Scheme The numerical scheme used in this work is the *Mixed Finite Element* U-V-W-P technique in which both velocity and pressure in the governing equations are regarded as primary variables and are discretized as unknowns. The first order Taylor-Galerkin method was used to discretize the equations.

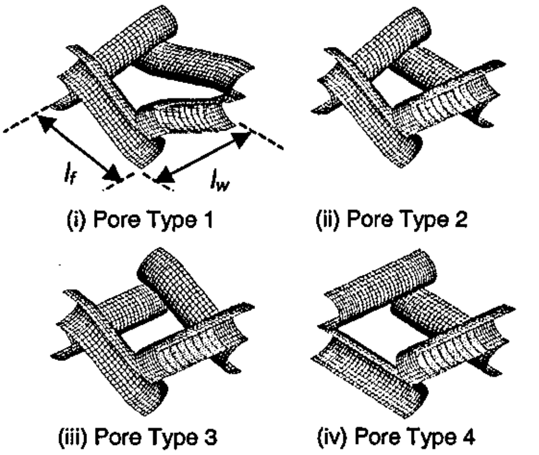
Boundary Conditions The velocity v_z corresponds to the approach velocity of the fluid towards the aperture and was set as 0.1 m s^{-1} at the inlet. The boundaries (the cloth filaments) were considered to be impermeable, non-slip surfaces where $v_x = v_y = v_z = 0$. A time step of Δt = 0.01 s was used throughout the simulations and the time stepping parameter θ used for the flow modelling was 0.95.

Physical Properties and Numerical Parameters A non-Newtonian fluid with the following properties was used in the simulations: density, ρ, 970 kg m^{-3}; consistency index, η_0, 80 kg m^{-1} s^{-1}; Power Law index, n, 0.90. Similar properties were used for the simulations of the Newtonian fluid, but with n taken to be 1.0.

An Intel Pentium IV 2.6 GHz processor was used to execute the computation to produce simulation results. The pre-processing work of the domain development and the mesh generation were handled using COSMOS GEOSTAR. Post-processing presentation of field variables such as flow velocity vectors and pressure contours were plotted using the graphic software package SURFER 8.

WOVEN MEDIA

The model can now be applied to the simulation of flow through a monofilament woven medium. The weaves considered in this work are the plain weave, the twill weave, and the satin weave. These give rise to the four types of basic pore shape that are shown in Figure 1, where only the half circle of each filament is illustrated.



(i) Pore Type 1 (ii) Pore Type 2

(iii) Pore Type 3 (iv) Pore Type 4

Fig. 1 Four types of basic weave patterns of a wire mesh.

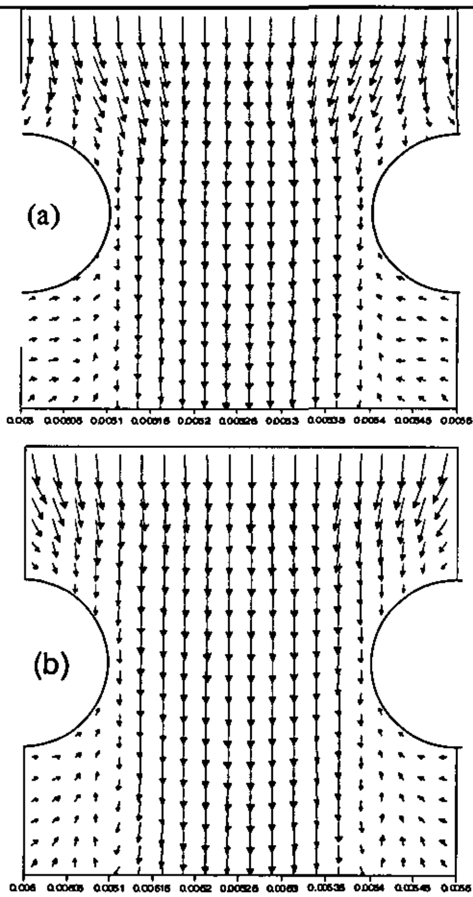


Fig. 2 Velocity vectors for flow of non-Newtonian fluid through pore type 1 at (a) *x-z* plane at $y = l_w/2$ (b) *y-z* plane at $x = l_f/2$.

RESULTS AND DISCUSSION

Fluid flow through a plain weave (pore type 1, Figure 1) was first investigated in conjunction with the experimental studies by Rushton (1969) using water and Chhabra and Richardson (1985) using shear thinning fluids. The ratio of aperture size to wire diameter used is 1.5 and the ratio of the domain length (*x*), width (*y*), height (*z*) and wire diameter (*d*) is 5:5:5:2. The flow inlet is at the top horizontal surface while the flow exit is at the lower horizontal surface.

A typical velocity vectors for the flow through wire mesh is illustrated in Figure 2. The flow is observed to swirl inwards on both sidewalls towards the centre on both *x-z* and *y-z* planes. This is consistent with the location of the wire mesh on both planes. Higher velocities were also observed at the area immediately above the wire mesh and in the section between the wires, compared to the velocity at the inlet and outlet.

The pressure profile illustrated in Figure 3 shows the expected high pressure on the surface of the wire meshes and decreasing rapidly as the fluid flows through the wire meshes. Cross section plots were presented for pressure contours in both the *x*-*z* plane at $y = l_w/2$ and the *y*-*z* plane at $x = l/2$.

The physical properties of the test fluids and the characteristic dimensions of the wire mesh and monofilament cloths used in the Chhabra and Richardson (1985) and Rushton (1969) experiments and in this study are summarized in Table 1.

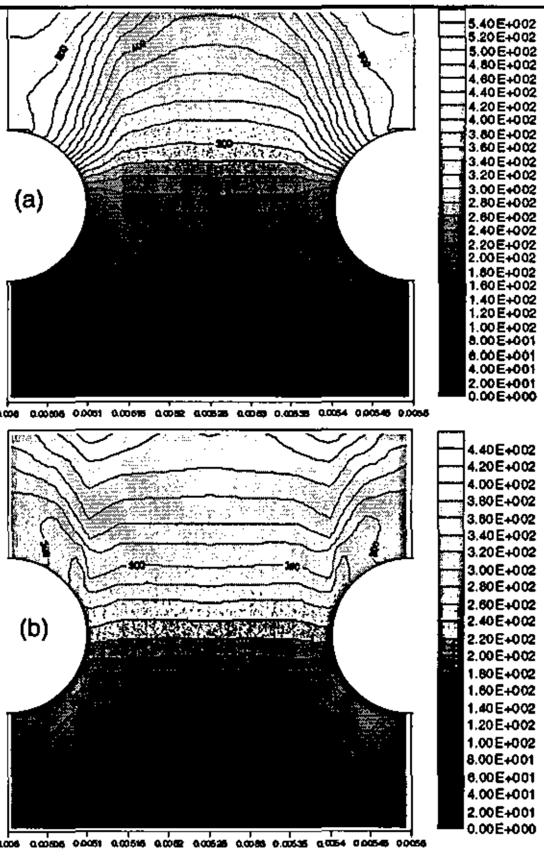


Fig. 3 Pressure contour for flow of non-Newtonian fluid through pore type 1 at (a) *x*-*z* plane at $y = l_w/2$ (b) *y*-*z* plane at $x = l/2$.

Chhabra and Richardson presented their experimental results for shear-thinning fluids using a loss coefficient C_D (which we refer to as a drag coefficient) defined as

$$C_D = \frac{2\Delta p}{\rho (u/\varepsilon)^2} \frac{1}{(1-\varepsilon)} \qquad (4)$$

where ε is the porosity of the wire mesh and Δp the pressure drop across the wire mesh. They also used a modified Reynolds number for a non-Newtonian fluid to $\mathrm{Re}_{NN} = \rho (u/\varepsilon)^{2-n} d^n / \eta_0$, a form similar to that used by Metzner (1956) and Skelland (1967) with porosity taken into consideration. In their paper, however, they erroneously took *d* to be the wire diameter rather than the hydraulic diameter, d_h, which can be taken as the aperture side dimension in the case of a square mesh. A more appropriate correlation for Re_{NN} consistent to that used by earlier researchers is therefore given as:

$$\mathrm{Re}_{NN} = \frac{\rho (u/\varepsilon)^{2-n} d_h^{\,n}}{\eta_0} \qquad (5)$$

where Re_{NN} reduces to the Reynolds number for a Newtonian liquid when $n = 1$. The experimental results presented in their paper were recalculated using equation (5) and presented in Figure 4. The results for water flow through monofilament nylon cloths reported by Rushton (1969) were also calculated using equations (4) and (5) and included on Figure 4. For the simulations in this work, the relationship between the flow rate and the pressure drop for each fluid is calculated from the model and subsequently presented using equations (4) and (5) and also plotted in Figure 5.

Table 1 Physical properties of the test fluids and characteristic dimensions of the plain weave wire mesh screens and monofilament cloth.

Test Liquid	n	η_0	Aperture	*d*	Symbols in
		(Pa s)	(μm)	(μm)	Figure 4
Chhabra and Richardson (1985); wire mesh screens					
1.5% CMC in water	0.60	6.6	53	36	■
1.5% CMC in water	0.59	7.6	53	36	◆
2.0% CMC in water	0.61	9.5	150	100	▲
2.5% CMC in water	0.34	45.0	150	100	●
Rushton (1969); monofilament Nylon cloth					
Water	1.0	0.001	24	30	+
	1.0	0.001	42.4	37	+
	1.0	0.001	60	30	+
	1.0	0.001	59	44	+
	1.0	0.001	71	61	+
	1.0	0.001	99	75	+
	1.0	0.001	144	105	+
	1.0	0.001	186	153	+
This study					
1.5% CMC in water	0.60	6.6	300	200	□
1.5% CMC in water	0.59	7.6	300	200	◇
2.0% CMC in water	0.61	9.5	300	200	△
2.5% CMC in water	0.34	45.0	300	200	○
Water	1.0	0.001	300	200	×

From Figure 4, the simulation results for water and the non-Newtonian liquid were observed to be in good agreement with the experimental results. Furthermore, the drag coefficient C_D and the Reynolds number Re_{NN} are related by a best fit curve of

$$C_D = \frac{10}{\mathrm{Re}_{NN}^{\;1.14}} \qquad (6)$$

The simulations were extended to pore types 2, 3 and 4; the drag coefficient C_D and the Reynolds number Re_{NN} for each pore type were calculated using equations (5) and (6) and plotted in Figure 5. From Figure 5, it is seen that pore type 1 gives the lowest pressure drop, while pore type 3 gives the highest pressure drop across the wire mesh with pore type 2, 4 in between them. This observation disagrees with the calculations by Lu *et al.* (1996) and Tung *et al.* (2002), who reported the lowest pressure drop for pore type 4 and highest pressure drop for pore type 1.

To explain this discrepancy, the experimental data of previous researchers and industrial wire mesh manufacturers were consulted. A closer examination of the experimental results reported by Armour and Cannon (1968) and Rushton and Griffiths (1971) showed a consistently lower pressure drop for plain weave compared to twilled weave, with satin weave ranging between them. As plain weave has the most uniform pore type amongst other weave patterns, where it consists entirely of pore type 1, plain weave can be safely used as a benchmark to check the accuracy of the numerical model.

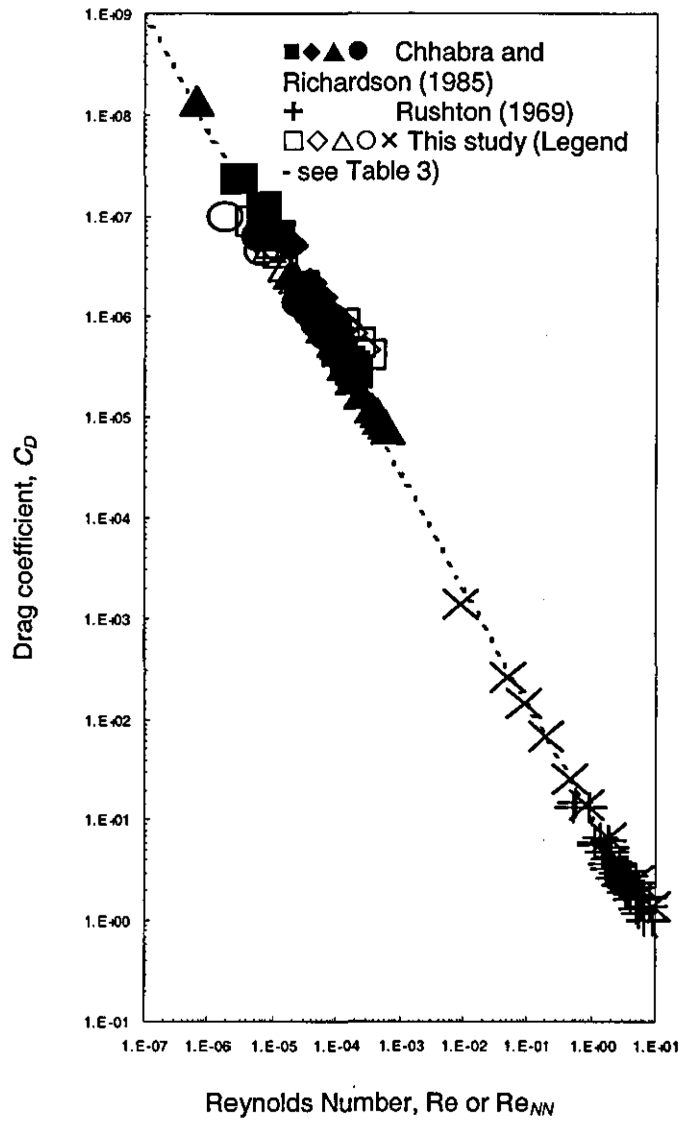


Fig. 4 Drag Coefficient *vs* Reynolds Number for Newtonian and non-Newtonian liquid flows through plain weave monofilament cloths and meshes.

The consistency of our results to that of Armour and Cannon (1968) and Rushton and Griffiths (1971) confirmed the accuracy of our predictions. Industrial woven cloth and wire mesh manufacturers have confirmed industrial experience whereby the plain weave is primarily used due to its lowest flow resistance - again confirming the validity of our results and giving confidence in our model.

CONCLUSION

A 3-D weighted residual finite element scheme for solving Stokes flow has been successfully developed and applied to simulate flow through monofilament wire meshes and fabrics. The accuracy of the model is proven by three test cases of increasing complexity and compared against experimental data for simulation of flow through wire meshes. The pressure drop across the wire mesh was analysed and the results were found to be in good agreement with the existing experimental literature. The results provide a sound basis to proceed with modelling of fluid flow through a wire mesh in order to quantify the effects of pressure drop, stress and particle capture in the domain of filters.

REFERENCES

Armour, J.C. and Cannon, J.N., 1968. Fluid flow through woven screens. *AIChEJ*, **14**(3): 415-420.

Chhabra, R. P. and Richardson, J. F., 1985. Flow of liquids through screens: Relationship between pressure drop and flow rate. *Chem. Eng. Sci.*, **40**(2): 313-316.

Ehrhardt, G., 1983. Flow measurements for wire gauzes. *Int. Chem. Eng.*, **23**:455-465.

Hanspal, N.S., Ruziwa, W.R., Nassehi, V. and Wakeman, R.J., 2004. Finite element modelling of flow of non-Newtonian fluids in pleated cartridge filters. *Proc. of 9th World Filtration Congress*, New Orleans.

Kiljanski, T. and Dziubinski, M., 1996. Resistance to flow of molten polymers through filtration screens. *Chem. Eng. Sci.*, **51**(19): 4533-4536.

Lu, W., Tung, K. and Hwang, K., 1996. Fluid flow through basic weaves of monofilament filter cloth. *Textile Res. J.*, **66**(5): 311-323.

Metzner, A. B., 1956. Non-Newtonian Technology : Fluid Mechanics, mixing, and heat transfer, in *Adv. Chem. Eng.*, Vol 1, edited by Drew T.B. and Hooper J.W. Academic Press, New York.

Nassehi, V., Hanspal, N.S., Waghode, A.N., Ruziwa, W.R. and Wakeman, R.J., 2005. Finite-element modelling of combined free/porous flow regimes: simulation of flow through pleated cartridge filters. *Chem. Eng. Sci.*, **60**(4): 995-1006.

Rushton, A., 1969. Filtration Research Report. Chem. Eng. Dept., UMIST.

Rushton, A. and Griffiths, P., 1971. Fluid flow in monofilament filter media. *Trans IChemE*, **49**: 49-59.

Ruziwa W.R., Hanspal N.S., Waghode A.N., Nassehi V. and Wakeman R.J., 2004. Computer modelling of pleated cartridge filters for viscous fluids. *Filtration*, **4**(2): 136-144.

Skelland, A. H. P., 1967. Non-Newtonian flow and heat transfer. Wiley, New York, USA.

Squiers, J.C., 1984. Fluid flow resistance models for wire weaves. *Filtration and Separation*, **10**:328-330.

Tung, K., Shiau, J., Chuang, C., Li, Y. and Lu, W., 2002. CFD analysis on fluid flow through multifilament woven filter cloths. *Sep. Sci. Technol.*, **37**(4): 799-821.

Weighardt, K.E.G., 1953. On the resistance of screens. *Aeronautical Quarterly*, **4**: 186-192.

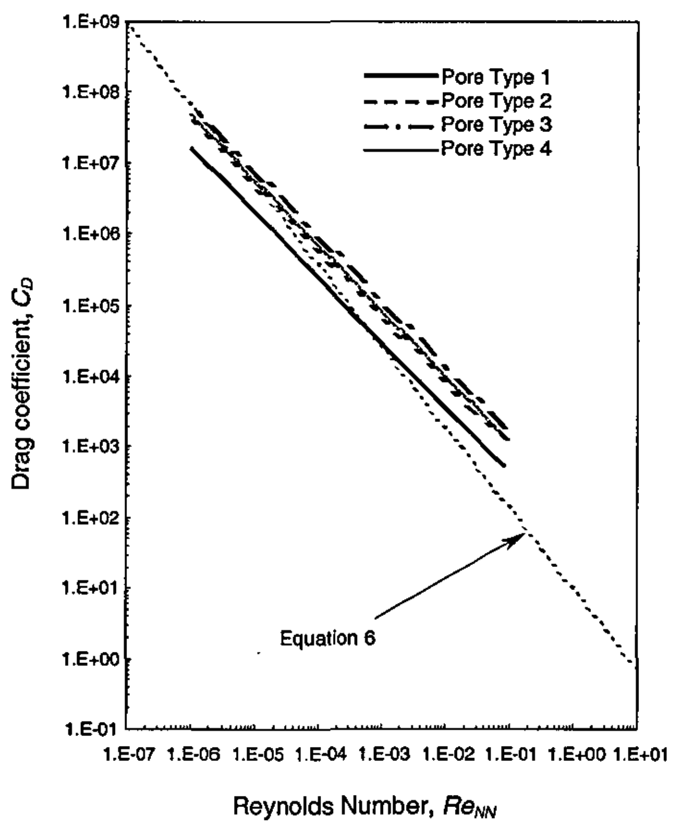


Fig. 5 Drag Coefficient *vs* Reynolds Number for non-Newtonian fluid flow through four basic pore types in monofilament cloths and meshes.

(4) TRANSACTIONS OF THE FILTRATION SOCIETY

Title: Modelling Flow in Monofilament Cloths :
I. Pressure Losses in Newtonian Fluids

Modelling Flow in Monofilament Cloths :
II. Pressure Losses in Non-Newtonian Fluids

Status: To be submitted.

Citation:

K. C. Ting, R. J. Wakeman and V. Nassehi. Modelling Flow in Monofilament Cloths : I. Pressure Losses in Newtonian Fluids, *Filtration.*

K. C. Ting, R. J. Wakeman and V. Nassehi. Modelling Flow in Monofilament Cloths : II. Pressure Losses in Non-Newtonian Fluids, *Filtration.*

APPENDIX 3

Program Manual

Introduction

This operating manual gives the details of the programs entitled 3DFEANOF-U version (3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow-UVWP version) and 3DFEANOF-P version (3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow-Penalty version). These codes were written in FORTRAN 90 programming language. An input file in ACSII format must be provided to run the program. The output file is RES.TXT - a text file containing the simulated results.

INPUT FILE FORMAT

3DFEANOF-U version (3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow-UVWP version)

The format of the input file is given with variables at each line explained.

Heading

Line 1 Format (A)

	TITLE	: Designated title of the input file

Master Data

Line 2 Format (2I5)

Variable 1	NCN	: Node per element
Variable 2	NGAUS	: Number of integration points

Line 3 Format (4I5)

Variable 3	NNP	: Number of nodes
Variable 4	NEL	: Number of elements
Variable 5	NBC	: Number of boundary conditions
Variable 6	NMAT	: Number of materials

Line 4 Format (2I5)

Variable 7	NTEP	: Results to be printed for every iterations if NTEP =1
Variable 8	ICORD	: For Cartesian coordinate system, icord=0 For cylindrical coordinate system, icord=1

Line 5 Format (3F10.0)

Variable 9	GRAV1	: Body force in *x*- direction
Variable 10	GRAV2	: Body force in *y*- direction
Variable 11	GRAV3	: Body force in *z*- direction

Line 6 Format (3F10.5)

Variable12	TOLV	: Convergence tolerance factor for pressure
Variable 13	TOLP	: Convergence tolerance factor for velocity
Variable 14	TOLC	: Convergence tolerance factor for variable to be added

Line 7 Format (9D10.5)

Variable15	RVISC	: Consistency coefficient
Variable 16	POWER	: Power Law index

Variable 17	TREF	: Reference temperature
Variable 18	TBCO	: Coefficient b in the power law model
Variable 19	TACO	: Coefficient a in the power law model
Variable 20	DISPC	: Coefficient for convective equation
Variable 21	PREF	: Reference pressure
Variable 22	RODEN	: Density
Variable 23	GAMAD	: Shear rate

Nodal Data

Line 8 – Line *m* Format (I7, 3E20.12)

Variable 24	M	: Node number M
Variable 25	X(M)	: X-coordinate of node M
Variable 26	Y(M)	: Y-coordinate of node M
Variable 27	Z(M)	: Z-coordinate of node M

Element Connectivity Data

Line *m* – Line *n* Format (21I7), with allocations for higher order elements

Variable 28	N	: Element number N
Variable 29	NODE(N, 1)	: Node number 1 of Element number N
Variable 30	NODE(N, 2)	: Node number 2 of Element number N
Variable 31	NODE(N, 3)	: Node number 3 of Element number N
Variable 32	NODE(N, 4)	: Node number 4 of Element number N
Variable 33	NODE(N, 5)	: Node number 5 of Element number N
Variable 34	NODE(N, 6)	: Node number 6 of Element number N
Variable 35	NODE(N, 7)	: Node number 7 of Element number N
Variable 36	NODE(N, 8)	: Node number 8 of Element number N

Boundary Condition Data

Line *n* – Line *k* Format (2I5, F10.4)

Variable 37	IBC	: Node number at which the boundary condition is applicable
Variable 38	JBC	: ‘1’ for *x*-direction velocity : ‘2’ for *y*-direction velocity : ‘3’ for *z*-direction velocity : ‘4’ for pressure
Variable 39	VBC	: Boundary condition value

INPUT FILE FORMAT

3DFEANOF-P version (3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow-Penalty version)

The format of the input file is given with variables at each line explained.

Heading

Line 1 Format (A)

	TITLE	: Designated title of the input file

Master Data

Line 2 Format (2I5)

Variable 1	NCN	: Node per element
Variable 2	NGAUS	: Number of integration points

Line 3 Format (4I5)

Variable 3	NNP	: Number of nodes
Variable 4	NEL	: Number of elements
Variable 5	NBC	: Number of boundary conditions
Variable 6	NMAT	: Number of materials

Line 4 Format (2I5)

Variable 7	NTEP	: Results to be printed for every iterations if NTEP =1
Variable 8	ICORD	: For Cartesian coordinate system, icord=0 For cylindrical coordinate system, icord=1

Line 5 Format (3F10.0)

Variable 9	GRAV1	: Body force in *x*- direction
Variable 10	GRAV2	: Body force in *y*- direction
Variable 11	GRAV3	: Body force in *z*- direction

Line 6 Format (3F10.5)

Variable12	TOLV	: Convergence tolerance factor for pressure
Variable 13	TOLP	: Convergence tolerance factor for velocity
Variable 14	TOLC	: Convergence tolerance factor for variable to be added

Line 7 Format (10D10.5)

Variable15	RVISC	: Consistency coefficient
Variable 16	POWER	: Power Law index

Variable 17	TREF	: Reference temperature
Variable 18	TBCO	: Coefficient b in the power law model
Variable 19	TACO	: Coefficient a in the power law model
Variable 20	DISPC	: Coefficient for convective equation
Variable 21	PREF	: Reference pressure
Variable 22	RODEN	: Density
Variable 23	GAMAD	: Shear rate
Variable 23a	PENAL	: Penalty parameter

Nodal Data

Line 8 – Line *m* Format (I7, 3E20.12)

Variable 24	M	: Node number M
Variable 25	X(M)	: X-coordinate of node M
Variable 26	Y(M)	: Y-coordinate of node M
Variable 27	Z(M)	: Z-coordinate of node M

Element Connectivity Data

Line *m* – Line *n* Format (21I7), with allocations for higher order elements

Variable 28	N	: Element number N
Variable 29	NODE(N, 1)	: Node number 1 of Element number N
Variable 30	NODE(N, 2)	: Node number 2 of Element number N
Variable 31	NODE(N, 3)	: Node number 3 of Element number N
Variable 32	NODE(N, 4)	: Node number 4 of Element number N
Variable 33	NODE(N, 5)	: Node number 5 of Element number N
Variable 34	NODE(N, 6)	: Node number 6 of Element number N
Variable 35	NODE(N, 7)	: Node number 7 of Element number N
Variable 36	NODE(N, 8)	: Node number 8 of Element number N

Boundary Condition Data

Line *n* – Line *k* Format (2I5, F10.4)

Variable 37	IBC	: Node number at which the boundary condition is applicable
Variable 38	JBC	: ‘1’ for *x*-direction velocity
		: ‘2’ for *y*-direction velocity
		: ‘3’ for *z*-direction velocity
		: ‘4’ for pressure
Variable 39	VBC	: Boundary condition value

APPENDIX 4

Program Listing

3DFEANOF

– U version

3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow – UVWP version

Fortran Program Source Code

1. Main Program Code

2. Subroutines

Kee Chien Ting

Advanced Separation Techniques Group
Department of Chemical Engineering
Loughborough University

```
c    ===================================================================
c
c    This is a program for the solution of non-newtonian, isothermal,
c    incompressible flow problems using the weighted residual galerkin
c    finite element method
c
c    The solution scheme is based on the U-V-W-P method
c
c    Velocity components and pressure are the prime unknowns in the flow
c    field.
c
c    This program uses eight-noded linear prism element.
c
c    Algebraic equations are solved by a frontal method.
c
c    A complete list of options is given on the program listing.
c
c    The program consists of a main module and subroutines
c
c    The program is written in FORTRAN programming language
c
c    This program is developed by Kee Chien Ting (last revised Nov 2004)
c
c    ===================================================================
c
c    work files
c    ==========
c      unit      contents
c ====================================================================
c       51     i     input data file
c              i
c       60     i     output file for documentation
c              i
c       11     i     output file containing velocity field data for
c              i     plotting
c              i
c       14     i     used as a work file in the solver routine
c              i
c       15     i     stores shape functions and their derivatives at
c              i     'full' integration points
c              i
c       17     i     output file containing pressure data for
c              i     contour plotting
c              i
c       20     i     output file containing elemental stiffness matrix
c              i     for element number 14 as seen on the mesh
c              i
c ====================================================================
c
c      List of variables
c      =================
c      aa   (   27,  27)   element coefficient matrices on LHS
c      K    (   27,  27)   element coefficient matrices on RHS
c      b    (    3,  20)   global derivatives of shape functions
c      bc   (maxdf     )   nodal constraints (boundary conditions)
c      cord (maxnp,ndim)   nodal coordinates
c      del  (    3,  20)   local  derivatives of shape functions
c      vel  (maxdf     )   nodal velocities  (displacements)
c      dsc1, dsc2          depths of slip layers
c      grav1               first  component of the applied body force
c      grav2               second component of the applied body force
c      icord               indicates whether the coordinate system is cartesian (planar) or
c                          cylindrical (axisymmetric)
c      tolp                convergence tolerance factor for pressures
c      tolv                convergence tolerance factor for velocities
c      nbc                 total number of boundary-node constraints
c      ncn                 number of nodes per element
c      ndf                 degree of freedom per node
c      ndim                dimensions of the solution domain
c      nel                 total number of elements
c      ngaus               number of integration points
c      nnp                 total number of nodal points
c      node (maxel,maxst)   element connectivity
c      nter                maximum number of iterations for non-newtonian case
c      num                 number of integration points per element
c      p    (    20    )   shape functions
c      press(maxnp     )   nodal pressures
c      r1   (maxdf     )   global load vector  (r.h.s.)
c      rfrct               friction coefficient (slip)
```

```
c      rr   (   27     )   element load vector
c      stiff(maxar     )   global stiffness matrix ( a in ax=r.h.s.)
c      rvisc               mu nought;consistency coefficient in power-law model
c      power               power law index
c      stemp               temperature
c      rtem                reference temperature
c      spress              pressure
c      rpress              reference pressure
c      tco                 coefficient relating viscosity to temperature
c      pco                 coefficient relating viscosity to pressure
c      gamad               shear rate
c      nwr                 no. of sample nodes for recording transient solutions
c
c      ====================================================================
c      List of Subroutines
c      ===================
c
c      bacsub              backsubstitution method for finding the final
c                          solution vector
c      clean               cleans the arrays and prepares them for
c                          solution
c      contol              makes a check for the convergence
c      deriv               calculates the jacobian matrix, its determinant
c                          and global derivatives of the shape functions
c      flow                calculates the velocities and pressures
c      front               frontal method for solving the final set
c                          of equations
c      gaussp              specifies the gauss points and weights for
c                          quadrature integration
c      getbcd              specifies the primary boundary conditions
c      getelm              specifies the nodal connectivity array
c      getmat              reads the input material data
c      getnod              reads the nodal co-ordinates for cartesian
c                          and axisymmetric systems
c      minimax             captures the minimum and maximum value for
c                          each specified variables
c      output              prints the final solution
c      putbcv              imposes the primary boundary conditions for
c                          velocity
c      secinv              calculates the second invariants
c      setprm              Sets the location data for nodal degrees of
c                          freedom
c      shape               calculates the shape functions and their
c                          derivatives
c      stress              calculates stress components at integration
c                          points
c      visca               calculates the viscosity
c
c     ==================================================================
       parameter (maxel  = 30000    )
       parameter (maxnp  = 37000    )
       parameter (maxbc  = 20000    )
       parameter (maxdf  = maxnp*4  )
       parameter (maxst  = 80       )
       parameter (maxfr  = 2000     )
       parameter (ndim   = 3        )

       implicit real*8 (a-h,o-z)
c
c     Storage allocation
c     ==================
c
       dimension title (          80)
       dimension node  (maxel,maxst) ,pmat (maxel,  9) ,cord (maxnp,ndim)
       dimension ncod  (maxdf      ), bc   (maxdf    )
       dimension ibc   (maxbc      ) ,jbc  (maxbc    ) ,vbc  (maxbc    )
       dimension vel   (maxdf      ) ,press(maxnp    )
       dimension r1    (maxdf      )
       dimension clump (maxnp      ) ,stres(maxnp,  6)
       dimension vet   (maxdf      ) ,cet  (maxnp    ) ,pet  (maxnp    )
       dimension nopp  (maxdf      )
       dimension aa    (maxst,maxst) ,rr   (maxst    )
       dimension xg    (          3) ,cg   (        3)
       dimension p     (         20) ,del  (    3, 20) ,b    (    3,  20)
       dimension ldest (maxst      ) ,kdest(maxst    ) ,nk   (maxst    )
       dimension eq    (maxfr,maxfr) ,lhed (maxfr    ) ,khed (maxfr    )
       dimension kpiv  (maxfr      ) ,lpiv (maxfr    ) ,jmod (maxfr    )
       dimension qq    (maxfr      ) ,pvkol(maxfr    ) ,sinv (maxel, 27)
       dimension mdf   (maxdf      ) ,ndn  (maxdf    )
       dimension ldsc  (22         )
       dimension temp  (maxnp      )
```

```fortran
       character *20 filnam
c
c     Opening of input and output data files
c     ======================================
c
       print*,'enter the name of your data file'
       read(*,2000) filnam

       open(unit=51,File=filnam,access='sequential',form='formatted',
     1      status="unknown",iostat=ios)

       open(unit=60,file='res.txt',access='sequential',form='formatted',
     1      status="unknown",iostat=ios)

       open(unit=17,file='stress.txt',access='sequential',form='formatted',
     1      status="unknown",iostat=ios)


       open(unit=14,form='unformatted',status='scratch',iostat=ios)
       open(unit=15,form='unformatted',status='scratch',iostat=ios)

         if(ios==0)then
               print*,"files opened"
         else
               print*,"files not opened"
         stop
         end if

       rewind 51
       rewind 60
       rewind 20
c
c      Initialize arrays
c     =================

       do 5010 itl = 1,maxel
       do 5010 ivl = 1,80
                           node (itl,ivl) = 0
5010   continue
       do 5020 itl = 1,maxel
       do 5020 ivl = 1,8
                           pmat (itl,ivl) = 0.0
5020   continue
       do 5030 itl = 1,maxnp
       do 5030 ivl = 1,3
                           cord (itl,ivl) = 0.0
5030   continue
       do 5040 itl = 1,maxnp
       do 5040 ivl = 1,6
                           stres(itl,ivl) = 0.0
5040   continue
       do 5050 itl = 1,maxdf
                           vel  (itl    ) = 0.0
5050   continue
       do 5060 itl = 1,maxdf
                           ncod   (itl)    = 0
                           rl     (itl)    = 0.0
                           bc     (itl)    = 0.0
                           vet    (itl)    = 0.0
                           mdf    (itl)    = 0
                           ndn    (itl)    = 0
                           nopp   (itl)    = 0
5060   continue
       do 5070 itl = 1,maxnp
                           clump  (itl)    = 0.0
                           cet    (itl)    = 0.0
                           pet    (itl)    = 0.0
                           press  (itl)    = 0.0
5070   continue
       do 5080 itl = 1,maxbc
                           ibc   (itl)     = 0
                           jbc   (itl)     = 0
                           vbc   (itl)     = 0.0
5080   continue
       do 5090 itl = 1,20
                           del   (1,itl)   = 0.0
                           del   (2,itl)   = 0.0
                           del   (3,itl)   = 0.0
5090   continue
       do 5100 itl = 1,maxst
```

```fortran
                           kdest (itl  )  = 0
                           nk    (itl  )  = 0
5100   continue
       do 5110 itl = 1,maxfr
                           lhed  (itl  )  = 0
                           khed  (itl  )  = 0
                           kpiv  (itl  )  = 0
                           lpiv  (itl  )  = 0
                           jmod  (itl  )  = 0
                           qq    (itl  )  = 0.0
                           pvkol (itl  )  = 0.0
       do 5110 ill = 1,maxfr
                           eq    (itl,ill)= 0.0
5110   continue
c-------------------------------------------------------------------------

c      Title of the program
c      ====================

       if(.not. eof(51)) read (51,2010) title
       write(60,4010) title

c      Element description data
c      ========================

       if (.not. eof(51)) read (51,2020) ncn ,ngaus
       print*, "ncn, ngaus read"
       write(60,4020) ncn ,ngaus

c      Mesh, boundary condition and material parameters
c      ================================================

       if (.not. eof(51)) read (51,2030) nnp  ,nel  ,nbc  ,nmat
       print*, "nnp, nel ,nbc ,nmat read"
       if (.not. eof(51)) read (51,2040) ntep ,icord
       if(icord.eq.0) write(60,4030)
       if(icord.eq.1) write(60,4040)

       write(60,4050)

       if(ntep.eq.0) ntep=1

c     ============================================================
c     icord=0 coordinate system is cartesian    (   planar   )
c     icord=1 coordinate system is cylindrical (axisymmetric)
c
c     if ntep = 1 then computed result after every iteration will
c     be printed ;if you do not need the result of intermediate
c     computations choose your own ntep;the result of first and
c     converged solutions will always be printed.
c     ============================================================

                if(nnp   .eq.0  .or.nnp  .gt.maxnp) then
                                         write(60,4060)
            elseif(nel   .eq.0  .or.nel  .gt.maxel) then
                                         write(60,4060)
            elseif(nbc   .eq.0  .or.nbc  .gt.maxbc) then
                                         write(60,4060)
            elseif(nmat  .eq.0  .or.nmat .gt.maxel) then
                                         write(60,4060)

            print*, 'the program is aborted"
            stop

                endif

       write(60,4070) nnp ,nel ,nbc ,nmat

       if (.not. eof(51)) read (51,2050) grav1, grav2, grav3
       print*, "grav1 grav2 grav3 read"
       write(60,4080) grav1, grav2, grav3

       if (.not. eof(51)) read(51,2060) tolv ,tolp, tolc
       print*, "tolv, tolp, tolc read"

       maxer=maxel

c     ==========================================================================
c     Read input data from main data file and prepare arrays for solution process
c     ==========================================================================
       call getmat(nel,nmat,pmat,51,60,maxel,rtem,rpef)
       call getnod(nnp,cord,51,60,maxnp,ndim,icord)
       call getelm(nel,ncn,node,51,60,maxer)
       call getbcd(nbc,ibc,jbc,vbc,51,60,maxbc)


c     ============================================================================
c                              Start of the time loop
c     ============================================================================
c
c      Set control parameters (default values are overwritten by input data
c      if specified)
c
c      ncn      number of nodes per element
c      ngaus    number of integration points
c      nter     maximum number of iterations for non-newtonian case
c      ndim     number of space dimensions in the solution domain
c
c       nter  = 5

                                           do 5130 ivel= 1,maxdf
                                           vel (ivel)  =   0.0
5130                                       continue
                                           do 5140 item= 1,maxnp
                                           temp(item)  =    rtem
5140                                       continue

c     ================================================================================
c                                     Transient data
c     ================================================================================
c
c      stime        starting time
c
c      deltat       time increment
c
c      alpha        indicates the choice of method being employed in alpha
c                   time stepping technique (backward difference,
c                   forward difference, central difference, galerkin)
c
c      nter         maximum number of time steps being employed for finding solution
c      ==============================================================================

       print*,"  "
       print*,"Enter the number of time steps desired"
       read*, nter
       write(60,4100) nter

       print*,"Enter the delta t desired"
       read*, deltat
       write(60,4110) deltat

       print*,"Type Code for taylor galerkin technique being used"
       print*,"Code 1:Forward Diference Method"
       print*,"Code 2:Galerkin's Method"
       print*,"Code 3:Central Diference Method"
       print*,"Code 4:Backward Diference Method"
       print*,"Code 5:Temporal Upwinding"

       read*, Code

           if(code .eq. 1) then
           alpha=0.0
           else if(code .eq. 2) then
           alpha=0.5
           else if(code .eq. 3) then
           alpha=2.0/3.0
           else if(code .eq. 4) then
           alpha=1.0
           else if(code .eq. 5) then
           print*,"Type in the value of alpha between 0.5 & 1"
           read *, alpha
           else
           print*, "Error in Typing code"
           end if

       print*, "alpha=",alpha

       do 5150 iter = 1 ,nter
       print*,'iter=',iter
       time = iter*deltat
```

```
       write(60,4090) iter

c      ================================================================================
c                             Calculate Nodal Velocities & Pressures
c      ================================================================================
       icho=1
                                                            rewind 11
                                                            rewind 14
                                                            rewind 15

                                               ndf    = 4
                                               ntov   = ndf * nnp
                                               ntrix  = ndf * ncn
       call clean
      1     (ncn   ,nel   ,ndf   ,node ,r1    ,maxel,maxst,maxdf,
      2      bc    ,ncod ,icho )

       call setprm
      1     (nnp   ,nel   ,ncn   ,node ,ndf   ,maxel,maxst,ndn   ,ntrix,
      2      maxdf,ntov ,mdf   ,nopp )

       call putbcv
      1   (nnp ,nbc ,ibc ,jbc ,vbc ,ncod ,bc,maxbc,maxdf,maxel,maxst,
      2    node)


c      idv4 is the file specifier for unit=20
c      ===================================

       idv4=20

       do 5160 iel=1,nel

       call flow (node ,cord ,pmat ,nopp ,mdf ,ndn ,ncod ,bc ,vel
      1,press, r1, temp,ldest,kdest,nk ,eq ,lhed ,khed ,kpiv ,lpiv,
      2jmod, qq, pvkol,iter ,nel ,ncn , ngaus,grav1 ,grav2, grav3, p,
      3del, b, ntrix, maxel, maxnp, maxst, maxfr, maxdf, ndim ,
      4aa ,xg ,da ,ntov ,num, icord, rr, iel, del1,deltat, alpha,idv4,
      5sinv, icho, nnp, tref)

5160   end do

c      ================================================================================
c                   calculates the second invariant of rate of deformation
c                                  tensor at integration points.
c      ================================================================================
       call secinv
      1    (nel   ,nnp   ,ncn   ,ngaus,node ,sinv ,cord ,p   ,b,
      2     del   ,da    ,vel   ,maxnp,maxel,maxst,ndim ,icord,
      3     maxdf,num)

c      ==============================================================
c                            Convergence check
c      ==============================================================
       call contol(vel ,iter ,ntov ,nnp,maxnp,maxdf,errov, errop
      1,vet ,cet, pet, press)

c      ==============================================================
c                        calculation of the nodal stress
c      ==============================================================
       call stress
      1      (nel,nnp,ncn ,node ,p  ,  b  ,  da  ,vel   ,maxnp, maxel,  maxst ,
      2       maxdf, stres, press, rvisc ,clump ,ngaus  )


c      ================
c      Print the output
c      ================

       iiter=(iter/ntep)*ntep

       if(iter.eq.1.or.iiter.eq.iter) then

       call output
      1   (nnp   ,vel   ,press, maxdf,maxnp,icord, stres)
       endif

c      ===============================================================
c                               End of time loop
c      ===============================================================
5150  continue

      close(51)
      close(unit=60)
      close(unit=11)
      close(unit=14)
      close(unit=15)
      close(20)
c
c    ==============================================================
c                         Read statements
c    ==============================================================

2000  format(a)
2010  format(80a)
2020  format(2i5)
2030  format(4i5)
2040  format(2i5)
2050  format(3f10.0)
2060  format(3f10.5)

c    ==============================================================
c                         Write statements
c    ==============================================================

4010  format(' ',5(/),' ',20x,60('*'),/' ',20x,'*',58x,'*',/
     1' ',20x,'*','  A  three  dimensional finite element model of a  ',
     29x,'*',/' ',20x,'*','  non-newtonian isothermal flow using ',
     320x,'*',/' ',20x,'*','  the UVP method.  ',39x,'*',/' ',20x,'*',
     558x,'*',/' ',20x,60('*')///,' ',20x,80('-'),/' ',20x,80a,/' ',
     620x,80('-'),///)

4020  format(' ',20x,3('['),' element description data',10('.'),/
     125x,'no.of nodes per element                  =',i10,/
     225x,'no.of integration points                 =',i10,/
     3//)

4030  format('    *** coordinate system is cartesian (planar) ***')
4040  format('*** coordinate system is cylindrical(axisymmetric) ***')
4050  format(' ')
4060  format(' ',10('['),'input data unacceptable',10(']')///)

4070  format(' ',20x,3('['),' mesh description data ',10('.'),/
     125x,'no.of nodal points                       =',i10,/
     225x,'no.of elements                           =',i10,/
     325x,'no.of nodal constraints on boundary      =',i10,/
     425x,'no.of different materials                =',i10,//)

4080  format(' ',20x,3('['),' uniform body force vector ',10('.'),/
     125x,'grav1                                    =',f15.4,/
     225x,'grav2                                    =',f15.4,/
     325x,'grav3                                    =',f15.4,//)
4090  format(///'     iteration no.',i5,//)
4100  format(///' Total number of time steps  =',i5,//)
4110  format(///' Deltat                      =',f15.4,//)


c     ====================================================================
      end program
c     ====================================================================


c     ==============================================================
      subroutine bacsub
     1    (ntotl,ifix ,vfix ,rhs  ,soln ,soln1, mfrnt,rwork,iwork,idv2,
     2     icho )

      implicit double precision(a-h,o-z)
      dimension ifix (ntotl),vfix (ntotl),rhs (ntotl),soln (ntotl)
      dimension rwork(mfrnt) ,iwork(mfrnt) ,soln1(ntotl)

                        do 6010 ipos=1,ntotl
                        soln(ipos) =0.0
                        if(ifix(ipos).ne.0) soln(ipos)=vfix(ipos)
6010                    continue


      do 6020 kpos=1,ntotl
      backspace idv2
```

```
      read(idv2)      ipos,ifrnt,jfrnt,(iwork(k),rwork(k),k=1,ifrnt)
      backspace idv2

      if(ifix(ipos).ne.0)  go to 6020

      ww            = 0.0
      rwork(jfrnt) = 0.0

                        do 6030 k=1,ifrnt
                        jpos=iabs(iwork(k))
                        ww  =ww - rwork(k)*soln(jpos)
6030                    continue

      soln (ipos)=rhs(ipos)+ww

6020  continue

      if (icho .eq. 2) goto 6050

      do  6040 ipos  = ((3*ntotl)/4)+1 , ntotl
              j      = ipos -((3*ntotl)/4)
            soln1(j) = soln(ipos)

6040  continue

6050  continue

      return
      end
c
c    =============================================================
      subroutine clean
     1    (ncn  ,nel  ,ndf  ,node ,r1   ,maxel,maxst,maxdf,
     2     bc   ,ncod ,icho )

      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     all arguments are defined elsewhere.
c
      dimension r1   (maxdf) ,node(maxel,maxst)
      dimension bc   (maxdf) ,ncod(maxdf      )
c
c     function
c     ========
c     cleans the used arrays and makes them ready for  solution
c
           do  6010 i    = 1,maxdf
                  r1(i) = 0.0
                  bc(i) = 0.0
                ncod(i) = 0
6010                   continue
                ntrix   = ndf *ncn
           do  6020 iel = 1,nel
           do  6020 inp = 1,ntrix
           node(iel,inp) = iabs(node(iel,inp))
6020                   continue
           if(icho.ne.1)then
           do 6030 iel = 1,nel
           write(11,3010)iel,(node(iel,j),j=1,ncn)
3010       format(10i5)
6030       continue

           endif

      return
      end
c
c    =============================================================
      subroutine contol
     1(vel  ,iter ,ntov ,nnp  ,maxnp,maxdf,errov,errop,vet ,cet,
     2 pet, press)

      implicit double precision(a-h,o-z)

      dimension vel   (maxdf),press(maxnp)
      dimension vet   (maxdf),cet  (maxnp), pet  (maxnp)

      errv  = 0.0
      torv  = 0.0
      torv  = 0.0
      errp  = 0.0
      torp  = 0.0
c
c     calculate difference between velocities in consecutive iterations
c     ==============================================================
c
                        do 6010 icheck = 1,ntov
      if(iter.eq.1) vet(icheck) = 0.0
                        errv = errv +
     1           (vel(icheck)-vet(icheck))  * (vel(icheck)-vet(icheck))
                        torv = torv + vel(icheck)*vel(icheck)

      vet(icheck) = vel(icheck)

6010                    continue
                        errov= errv/torv

c
c      calculate difference between pressures in consecutive iterations
c      ==============================================================

                        do 6030 icheck = 1,nnp
      if(iter.eq.1)   pet(icheck) = 0.0
                              errp = errp +
     1        (press(icheck)-pet(icheck))*(press(icheck)-pet(icheck))
                        torp = torp + press(icheck)*press(icheck)

      pet(icheck) = press(icheck)

6030                    continue
                        errop= errp/torp
      return
      end

c      ================================================================
      subroutine deriv
     1    (iel  ,ig   ,jg   ,kg, p    ,del  ,b ,ncn ,da ,cg ,node,
     2     cord ,maxel,maxnp)

      implicit double precision(a-h,o-z)
      dimension p(20),b(3,40),del(3,20),cg(3),cj(3,3),cji(3,3)
      dimension node(maxel,27),cord(maxnp,3)

      do 6010 j=1,3
      do 6010 l=1,3
      gash=0.0
      do 6020 k=1,ncn
      nn=iabs(node(iel,k))

6020  gash=gash + del(j,k)*cord(nn,l)

      cj(j,l)=gash

6010  continue

      detj =  cj(1,1)*cj(2,2)*cj(3,3)+cj(2,1)*cj(3,2)*cj(1,3)
     1      + cj(1,2)*cj(2,3)*cj(3,1)-cj(1,3)*cj(2,2)*cj(3,1)
     2      - cj(1,2)*cj(2,1)*cj(3,3)-cj(2,3)*cj(3,2)*cj(1,1)

      if(detj.le.0.0) then
      write(60,3010) iel,detj
3010  format(1x ,' Error: Zero or Negative Jacobian. ', i6,g20.5)
      stop
      endif

      cji(1,1) =  (cj(2,2)*cj(3,3)-cj(3,2)*cj(2,3))  / detj
      cji(1,2) = ((cj(1,2)*cj(3,3)-cj(3,2)*cj(1,3))) / detj
      cji(1,3) =  (cj(1,2)*cj(2,3)-cj(2,2)*cj(1,3))  / detj
      cji(2,1) = ((cj(2,1)*cj(3,3)-cj(3,1)*cj(2,3))) / detj
      cji(2,2) =  (cj(1,1)*cj(3,3)-cj(3,1)*cj(1,3))  / detj
      cji(2,3) = ((cj(1,1)*cj(2,3)-cj(2,1)*cj(1,3))) / detj
      cji(3,1) =  (cj(2,1)*cj(3,2)-cj(3,1)*cj(2,2))  / detj
      cji(3,2) = ((cj(1,1)*cj(3,2)-cj(3,1)*cj(1,2))) / detj
      cji(3,3) =  (cj(1,1)*cj(2,2)-cj(2,1)*cj(1,2))  / detj

      do 6030 j=1,3
      do 6030 l=1,ncn
      b(j,l)=0.0
      do 6030 k=1,3
```

```
      da= detj*cg(ig)*cg(jg)*cg(kg)

      return
      end

c     ================================================================
      subroutine flow(node ,cord ,pmat ,nopp ,mdf ,ndn ,ncod ,bc ,vel
     1,press, r1, temp,ldest,kdest,nk ,eq ,lhed ,khed ,kpiv ,lpiv,
     2jmod, qq, pvkol, iter ,nel ,ncn , ngaus,grav1,
     3grav2, grav3, p, del, b, ntrix, maxel, maxnp, maxst, maxfr, maxdf,
     4ndim ,aa ,xg ,da ,ntov ,num, icord, rr, iel, del1,deltat,alpha,
     5idv4,sinv, icho, nnp, tref)

      implicit double precision(a-h,o-z)

      dimension node (maxel,maxst),pmat (maxel,   9),cord (maxnp, ndim)
      dimension ncod (maxdf      ),bc   (maxdf      ),sinv (maxel,   27)
      dimension vel  (maxnp ,   3),r1   (maxdf      )
      dimension aa   (maxst,maxst),rr   (maxst      ),ldest(maxst      )
      dimension xg   (           3),cg   (          3),kdest(maxst      )
      dimension x    (           3),v    (          3),nk   (maxst      )
      dimension bicn (           2),hh   (          3)
      dimension p    (          20),del  (   3,   20),b    (   3,    20)
      dimension eq   (maxfr,maxfr),nopp (maxdf      )
      dimension ldsc (          22)
      dimension lhed (maxfr      ),khed (maxfr      ),jmod (maxfr      )
      dimension lpiv (maxfr      ),kpiv (maxfr      ),qq   (maxfr      )
      dimension pvkol(maxfr      ),mdf  (maxdf      ),ndn  (maxdf      )
      dimension ppp  (20    ,   20),pp   (20         )
      dimension ak   (100,100)
      dimension akf  (100    )
      dimension NQ   (20    ,   20),NP   (3    ,    4)
      dimension C    (maxst      ),temp (maxnp      )
      dimension DEL1 (3          )
      dimension press(maxnp      ),clump(maxnp      ),SHAPE1D(3        )
      dimension gdsf (    3,   20)
      dimension dmass(100, 100)

                                                       rvisc = pmat(iel,1)
                                                       rpef  = pmat(iel,2)
                                                       power = pmat(iel,3)
                                                       rtem  = pmat(iel,4)
                                                       tbco  = pmat(iel,5)
                                                       taco  = pmat(iel,6)
                                                       roden = pmat(iel,8)
                                                       gamad = pmat(iel,9)


      velsound = 1150.0
      beta     =    0.0
      permx    = 1.0E-5
      permy    = 1.0E-5
      permz    = 1.0E-5


                    do 6010 idf= 1,ntrix
                    rr (idf)       = 0.0
                    akf(idf)       = 0.0
                    C  (idf)       = 0.0
                    do 6010 jdf= 1,ntrix

                    aa    (idf,jdf)=0.0
                    dmass(idf,jdf)=0.0
                    ak    (idf,jdf)=0.0
6010                continue


      call gaussp(ngaus,xg,cg)

      lg=0

      do 6020 ig=1,ngaus
                                         g = xg(ig)
      do 6020 jg=1,ngaus
                                         h = xg(jg)
      do 6020 kg=1,ngaus
                                         f = xg(kg)
      if(iter.eq.1) then

      call shape (g,h,f,p,del,ncn)

      call deriv (iel,ig,jg,kg,p,del,b,ncn,da,cg,node,cord,
     1            maxel,maxnp)

                 iig=ig
                 jjg=jg
                 kkg=kg

                 write(15)   iel ,ig ,jg ,kg, p ,del ,b ,da
      else
      if(.not. EOF(15))read(15) iel,iig,jjg,kkg,p ,del ,b , da
      endif
c
c     calculation of viscosity based on the constitutive equation.
c
      spress = 0.0
      stemp  = 0.0

              do 5333 ip = 1,ncn
              jp    = iabs(node(iel,ip))
              stemp = stemp + temp(jp) * p(ip)
 5333         continue
              epsii = 1.d-10
              gamad = sinv(iel,lg)
              if(gamad.lt.epsii) gamad = epsii

      call visca
     1(rvisc,power,visc,stemp,rtem,tbco,spress,rpef,taco,gamad)

c
c     prepration of the convective acceleration terms/balancing
c     dissipation is used
c
                    do 6050 idff= 1,3
                    x(idff)      = 0.0
                    v(idff)      = 0.0
                    hh(idff)     = 0.0
6050                continue
         do 6060 icn = 1 ,ncn
              jcn = iabs(node(iel,icn))
         do 6060 idff= 1 ,   3
         x(idff)     = x(idff) + p(icn)*cord(jcn,idff)
         v(idff)     = v(idff) + p(icn)*vel (jcn,idff)
6060     continue


      if(icord.eq.1) then
c
c     modify da for axisymmetric computations.
c
      da = da * x(1)
      endif


c     column index

      do 6070 i=1,ncn

                     j11= i
                     j12= i + ncn
                     j13= i + 2*ncn
                     j14= i + 3*ncn

      do 6070 j=1,ncn
                     j21= j
                     j22= j + ncn
                     j23= j + 2*ncn
                     j24= j + 3*ncn


c     Dicretized form of 3D Stokes Equation
c
c --- Stiffness Matrix of Left Hand Side --------------------------------------------------
c
```

```
-     --- [illegible] [illegible] (direction of ordinate system)

      aa(j11,j21)=aa(j11,j21) +  p(i)*p(j)*da
     1                        +  alpha*deltat*((2.0*visc/roden)
     2                           +(0.5*deltat*velsound*velsound))
     3                           *b(1,i)*b(1,j)*da
     3                        +  alpha*deltat*(visc/roden)*b(2,i)
     4                           *b(2,j)*da
     5                        +  alpha*deltat*(visc/roden)*b(3,i)
     6                           *b(3,j)*da

      aa(j11,j22)=aa(j11,j22) +  alpha*deltat*(visc/roden)
     1                           *b(2,i)*b(1,j)*da
     2                        +  alpha*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(1,i)*b(2,j)*da

      aa(j11,j23)=aa(j11,j23) +  alpha*deltat*(visc/roden)*b(3,i)
     1                           *b(1,j)*da
     2                        +  alpha*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(1,i)*b(3,j)*da

      aa(j11,j24)=aa(j11,j24) -  (1.0/roden)*alpha*deltat*b(1,i)
     1                           *p(j)*da


      aa(j12,j21)=aa(j12,j21) +  alpha*deltat*(visc/roden)*b(1,i)
     1                           *b(2,j)*da
     2                        +  alpha*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(2,i)*b(1,j)*da

      aa(j12,j22)=aa(j12,j22) +  p(i)*p(j)*da
     1                        +  alpha*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     3                        +  alpha*deltat*((2.0*visc/roden)
     4                           +(0.5*deltat*velsound*velsound))
     5                           *b(2,i)*b(2,j)*da
     5                        +  alpha*deltat*(visc/roden)*b(3,i)
     6                           *b(3,j)*da


      aa(j12,j23)=aa(j12,j23) +  alpha*deltat*(visc/roden)*b(3,i)
     1                           *b(2,j)*da
     2                        +  alpha*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(2,i)*b(3,j)*da

      aa(j12,j24)=aa(j12,j24) -  (1.0/roden)*alpha*deltat*b(2,i)
     1                           *p(j)*da


      aa(j13,j21)=aa(j13,j21) +  alpha*deltat*(visc/roden)*b(1,i)
     1                           *b(3,j)*da
     2                        +  alpha*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(3,i)*b(1,j)*da


      aa(j13,j22)=aa(j13,j22) +  alpha*deltat*(visc/roden)*b(2,i)
     1                           *b(3,j)*da
     2                        +  alpha*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(3,i)*b(2,j)*da


      aa(j13,j23)=aa(j13,j23) +  p(i)*p(j)*da
     1                        +  alpha*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     3                        +  alpha*deltat*(visc/roden)*b(2,i)
     4                           *b(2,j)*da
     5                        +  alpha*deltat*((2.0*visc/roden)
     6                           +(0.5*deltat*velsound*velsound))
     7                           *b(3,i)*b(3,j)*da


      aa(j13,j24)=aa(j13,j24) -  (1.0/roden)*alpha*deltat*b(3,i)
     1                           *p(j)*da


      aa(j14,j21)=aa(j14,j21) -  roden*alpha*deltat*velsound*velsound
     1                           *p(i)*b(1,j)*da


      aa(j14,j22)=aa(j14,j22) -  roden*alpha*deltat*velsound*velsound
     1                           *p(i)*b(2,j)*da
```

```
      aa(j14,j23)=aa(j14,j23) -  roden*alpha*deltat*velsound*velsound
     1                           *p(i)*b(3,j)*da


      aa(j14,j24)=aa(j14,j24) -  p(i)*p(j)*da
     1                        -  (0.5*alpha*deltat)*roden*deltat
     2                           *velsound*velsound*b(1,i)*b(1,j)*da
     3                        -  (0.5*alpha*deltat)*roden*deltat
     4                           *velsound*velsound*b(2,i)*b(2,j)*da
     5                        -  (0.5*alpha*deltat)*roden*deltat
     6                           *velsound*velsound*b(3,i)*b(3,j)*da

c --- Matrix on Right Hand Side ---------------------------------------------
c
c     For Transient State (Cartesian co-ordinate system)

      ak(j11,j21)=ak(j11,j21) +  p(i)*p(j)*da
     1                        -  (1.0-alpha)*deltat*((2.0*visc/roden)
     2                           +(0.5*deltat*velsound*velsound))
     3                           *b(1,i)*b(1,j)*da
     4                        -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     5                           *b(2,j)*da
     6                        -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     7                           *b(3,j)*da


      ak(j11,j22)=ak(j11,j22) -  (1.0-alpha)*deltat*(visc/roden)
     1                           *b(2,i)*b(1,j)*da
     2                        -  (1.0-alpha)*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(1,i)*b(2,j)*da


      ak(j11,j23)=ak(j11,j23) -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     1                           *b(1,j)*da
     2                        -  (1.0-alpha)*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(1,i)*b(3,j)*da


      ak(j11,j24)=ak(j11,j24) +  (1.0/roden)*(1.0-alpha)*deltat*b(1,i)
     1                           *p(j)*da


      ak(j12,j21)=ak(j12,j21) -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     1                           *b(2,j)*da
     2                        -  (1.0-alpha)*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(2,i)*b(1,j)*da


      ak(j12,j22)=ak(j12,j22) +  p(i)*p(j)*da
     1                        -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     3                        -  (1.0-alpha)*deltat*((2.0*visc/roden)
     4                           +(0.5*deltat*velsound*velsound))
     5                           *b(2,i)*b(2,j)*da
     6                        -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     7                           *b(3,j)*da


      ak(j12,j23)=ak(j12,j23) -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     1                           *b(2,j)*da
     2                        -  (1.0-alpha)*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(2,i)*b(3,j)*da


      ak(j12,j24)=ak(j12,j24) +  (1.0/roden)*(1.0-alpha)*deltat*b(2,i)
     1                           *p(j)*da


      ak(j13,j21)=ak(j13,j21) -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     1                           *b(3,j)*da
     2                        -  (1.0-alpha)*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(3,i)*b(1,j)*da


      ak(j13,j22)=ak(j13,j22) -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     1                           *b(3,j)*da
     2                        -  (1.0-alpha)*deltat*(0.5*deltat*velsound
     3                           *velsound)*b(3,i)*b(2,j)*da
```

```
      ak(j13,j23)=ak(j13,j23) +  p(i)*p(j)*da
     1                        -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     3                        -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     4                           *b(2,j)*da
     5                        -  (1.0-alpha)*deltat*((2.0*visc/roden)
     6                           +(0.5*deltat*velsound*velsound))
     7                           *b(3,i)*b(3,j)*da


      ak(j13,j24)=ak(j13,j24) +  (1.0/roden)*(1.0-alpha)*deltat*b(3,i)
     1                           *p(j)*da


      ak(j14,j21)=ak(j14,j21) +  roden*(1.0-alpha)*deltat*velsound
     1                           *velsound*p(i)*b(1,j)*da


      ak(j14,j22)=ak(j14,j22) +  roden*(1.0-alpha)*deltat*velsound
     1                           *velsound*p(i)*b(2,j)*da


      ak(j14,j23)=ak(j14,j23) +  roden*(1.0-alpha)*deltat*velsound
     1                           *velsound*p(i)*b(3,j)*da


      ak(j14,j24)=ak(j13,j24) -  p(i)*p(j)*da
     1                        +  (0.5*deltat)*roden*(1.0-alpha)*deltat
     2                           *velsound*velsound*b(1,i)*b(1,j)*da
     3                        +  (0.5*deltat)*roden*(1.0-alpha)*deltat
     4                           *velsound*velsound*b(2,i)*b(2,j)*da
     5                        +  (0.5*deltat)*roden*(1.0-alpha)*deltat
     6                           *velsound*velsound*b(3,i)*b(3,j)*da

c
c     Body Force Effect (for Elemental Load Vector Calculation)
c     ==========================================================
c

      C(j11) =C(j11) + (1.0-alpha)*deltat*p(j)*grav1*da

      C(j12) =C(j12) + (1.0-alpha)*deltat*p(j)*grav2*da

      C(j13) =C(j13) + (1.0-alpha)*deltat*p(j)*grav3*da

      C(j14) =C(j14) + 0


6070  continue
6020  continue

c     For Transient State (Cartesian Co-ordinate System)
c     ==================================================
c     Term one on RHS is evaluated
c     ============================

      do 6080 i=1,ncn

                     j11= i
                     j12= i + ncn
                     j13= i + 2*ncn
                     j14= i + 3*ncn

      do 6080 j=1,ncn
                     j21= j
                     j22= j + ncn
                     j23= j + 2*ncn
                     j24= j + 3*ncn

      nn=iabs(node(iel,j))


      akf(j11)=akf(j11) + ak(j11,j21)*vel(nn,1) +
     1                    ak(j11,j22)*vel(nn,2) +
     2                    ak(j11,j23)*vel(nn,3) +
     3                    ak(j11,j24)*press(nn)


      akf(j12)=akf(j12) + ak(j12,j21)*vel(nn,1) +
     1                    ak(j12,j22)*vel(nn,2) +
     2                    ak(j12,j23)*vel(nn,3) +
       akf(j13)=akf(j13) + ak(j13,j21)*vel(nn,1) +
      1                    ak(j13,j22)*vel(nn,2) +
      2                    ak(j13,j23)*vel(nn,3) +
      3                    ak(j13,j24)*press(nn)


       akf(j14)=akf(j14) + ak(j14,j21)*vel(nn,1) +
      1                    ak(j14,j22)*vel(nn,2) +
      2                    ak(j14,j23)*vel(nn,3) +
      3                    ak(j14,j24)*press(nn)

 6080  continue

c
c     Evaluation of Elemental Load Vector
c     ===================================

       do 6085 i=1,ncn

                      j11= i
                      j12= i + ncn
                      j13= i + 2*ncn
                      j14= i + 3*ncn

c      For Transient State  (Cartesian Co-ordinate System)

       rr(j11)= rr(j11) + akf(j11) + C(j11)
       rr(j12)= rr(j12) + akf(j12) + C(j12)
       rr(j13)= rr(j13) + akf(j13) + C(j13)
       rr(j14)= rr(j14) + akf(j14) + C(j14)

 6085  continue

       maxte=maxdf
       call front
      1(aa    ,rr    ,iel   ,node ,maxel,maxst,ldest,kdest,nk    ,maxfr
      2,eq    ,lhed ,khed ,kpiv ,lpiv ,jmod ,qq    ,pvkol,vel   ,r1
      3,ncod ,bc    ,nopp ,mdf   ,ndn   ,maxdf,nel   ,maxte,ntov ,lcol
      4,nell ,ntra, press,icho )

       return
       end
c      ================================================================
       subroutine front
      1    (aa    ,rr    ,iel   ,nop   ,maxel,maxst,ldest,kdest,nk    ,maxfr,
      2     eq    ,lhed ,khed ,kpiv ,lpiv ,jmod ,qq    ,pvkol,vel   ,r1    ,
      3     ncod ,bc    ,nopp ,mdf   ,ndn   ,maxdf,nel   ,maxte,ntov ,lcol ,
      4     nell ,ntra, press,icho )
c
c      Frontal elimination routine using diagonal pivoting
c
       implicit double precision(a-h,o-z)
       dimension aa    (maxst,maxst) ,rr    (maxst)
       dimension nop   (maxel,maxst)
       dimension ldest(maxst)         ,kdest(maxst)        ,nk    (maxst)
       dimension eq    (maxfr,maxfr) ,lhed (maxfr)        ,khed (maxfr)
       dimension kpiv (maxfr)         ,lpiv (maxfr)
       dimension jmod (maxfr)         ,qq    (maxfr)        ,pvkol(maxfr)
       dimension vel   (maxte)         ,r1    (maxdf)        ,ncod (maxdf)
       dimension bc    (maxdf)         ,nopp (maxdf)        ,mdf   (maxdf)
       dimension ndn   (maxdf)         ,press(maxdf)
c
c      nlp and ndl are the file specifiers for units 60 and 14 respectively
c      ====================================================================

       nlp=60
       ndl=14
c
c      Prefront
c      ========
       nmax=maxfr
       ncrit=20
       nlarg=maxfr-10
       if(iel.eq.1) nell = 0
       if(iel.eq.1) ntra = 1
       if(ntra.eq.0) goto 6040
       nmax = maxfr
       ntra = 0
```

```
       lfron = 0
       nlarg = nmax-10
c
c      Find last appeareance of each node
c      ==================================
       nlast = 0
       do 6010 i = 1,ntov
       do 6020 n = 1,nel
       jdn = ndn(n)
       do 6030 l = 1,jdn
       if(nop(n,l).ne.i)go to 6030
       nlast1 = n
       nlast = n
       ll = l
 6030  continue
 6020  continue
       if(nlast.eq.0) go to 6010
       nop(nlast,ll) = -nop(nlast,ll)
       nlast = 0
 6010  continue
       ntrix = jdn
c
c      Assembly
c      ========
 6040  continue
       if(iel.gt.1) go to 6060
       lcol = 0
       do 6050 i = 1,nmax
       do 6050 j = 1,nmax
       eq(j,i) = 0.
 6050  continue
 6060  nell = nell+1
       n = nell
       jdn = ndn(nell)
       kc = 0
       do 6070 j = 1,jdn
       nn = nop(n,j)
       m = iabs(nn)
       k = nopp(m)
       idf = mdf(m)
       r1(m) = rr(j)+r1(m)
       do 6070 l = 1,idf
       kc = kc+1
       ii = k+l-1
       if(nn.lt.0)ii = -ii
       nk(kc) = ii
 6070  continue
c
c      Set up heading vectors
c      ======================
c
       do 6080 lk = 1,kc
       node = nk(lk)
       if(lcol.eq.0)goto 6100
       do 6090 l = 1,lcol
       ll = l
       if(iabs(node).eq.iabs(lhed(l)))go to 6110
 6090  continue
 6100  lcol = lcol+1
       ldest(lk) = lcol
       lhed(lcol) = node
       go to 6080
 6110  ldest(lk) = ll
       lhed(ll) = node
 6080  continue
       if(lcol.le.nmax)go to 6130
       nerror = 2
       write(nlp,3010)nerror
       stop
 6130  continue
       do 6140 l = 1,kc
       ll = ldest(l)
       do 6140 k = 1,kc
       kk = ldest(k)
       eq(kk,ll) = eq(kk,ll)+aa(k,l)
 6140  continue
       if(lcol.lt.ncrit.and.nell.lt.nel) return
c
c      Find out which matrix elements are fully assembeled
c      ===================================================
      ir = 0
      do 6160 l = 1,lcol
      kt = lhed(l)
      if(kt.ge.0)go to 6160
      lc = lc+1
      lpiv(lc) = l
      kro = iabs(kt)
      if(ncod(kro).ne.1)go to 6160
      ir = ir+1
      jmod(ir) = l
      ncod(kro) = 2
      r1(kro) = bc(kro)
6160  continue
c
c     Modify equations with applied boundary conditions
c     =================================================
      if(ir.eq.0)go to 6190
      do 6170 irr = 1,ir
      k = jmod(irr)
      kh = iabs(lhed(k))
      do 6180 l = 1,lcol
      eq(k,l) = 0.
      lh = iabs(lhed(l))
      if(lh.eq.kh)eq(k,l) = 1.
6180  continue
6170  continue
6190  continue
      if(lc.gt.0)go to 6200
      ncrit = ncrit+10
c     write(nlp,3020)ncrit
      if(ncrit.le.nlarg) return
      nerror = 3
      write(nlp,3030)nerror
      stop
6200  continue
c
c     Search for absolute pivot
c     =========================
      pivot = 0.
      do 6210 l = 1,lc
      lpivc = lpiv(l)
      kpivr = lpivc
      piva = eq(kpivr,lpivc)
      if(abs(piva).lt.abs(pivot))go to 6220
      pivot = piva
      lpivco = lpivc
      kpivro = kpivr
6220  continue
6210  continue
      if(pivot.eq.0.0) return
c
c     Normalise pivotal row
c     =====================
      lco = iabs(lhed(lpivco))
      kro = lco

c     if(nit.eq.0.or.npra.eq.0)go to 6230
c6230 continue

      if(abs(pivot).lt.0.1d-28) write(nlp,3050)
      do 6240 l = 1,lcol
      qq(l) = eq(kpivro,l)/pivot
6240  continue
      rhs = r1(kro)/pivot
      r1(kro) = rhs
      pvkol(kpivro) = pivot
c
c     Eliminate then delete pivotal row and column
c     ============================================
      if(kpivro.eq.1)go to 6300
      kpivr = kpivro-1
      do 6250 k = 1,kpivr
      krw = iabs(lhed(k))
      fac = eq(k,lpivco)
      pvkol(k) = fac
      if(lpivco.eq.1.or.fac.eq.0.)go to 6270
      lpivc = lpivco-1
      do 6260 l = 1,lpivc
      eq(k,l) = eq(k,l)-fac*qq(l)
6260  continue
```

```
      lpivc = lpivco+1
      do 6280 l = lpivc,lcol
      eq(k,l-1) = eq(k,l)-fac*qq(l)
6280  continue
6290  r1(krw) = r1(krw)-fac*rhs
6250  continue
6300  if(kpivro.eq.lcol)go to 6360
      kpivr = kpivro+1
      do 6310 k = kpivr,lcol
      krw = iabs(lhed(k))
      fac = eq(k,lpivco)
      pvkol(k) = fac
      if(lpivco.eq.1)go to 6330
      lpivc = lpivco-1
      do 6320 l = 1,lpivc
      eq(k-1,l) = eq(k,l)-fac*qq(l)
6320  continue
6330  if(lpivco.eq.lcol)go to 6350
      lpivc = lpivco+1
      do 6340 l = lpivc,lcol
      eq(k-1,l-1) = eq(k,l)-fac*qq(l)
6340  continue
6350  r1(krw) = r1(krw)-fac*rhs
6310  continue
6360  continue
c
c     Write pivotal equation on disc
c     ==============================
      write(nd1) kro,lcol,lpivco,(lhed(l),qq(l),l = 1,lcol)
      do 6370 l = 1,lcol
      eq(l,lcol) = 0.
      eq(lcol,l) = 0.
6370  continue
c
c     Rearrange heading vectors
c     =========================
      lcol = lcol-1
      if(lpivco.eq.lcol+1)go to 6390
      do 6380 l = lpivco,lcol
      lhed(l) = lhed(l+1)
6380  continue
6390  continue
c
c     Determine whether to assemble,eliminate,or backsubstitute
c     =========================================================
      if(lcol.gt.ncrit)go to 6150
      if(nell.lt.nel) return
      if(lcol.gt.1)go to 6150
      lco = iabs(lhed(1))
      kpivro = 1
      pivot = eq(1,1)
      kro = lco
      lpivco = 1
      qq(1) = 1.

c     if(nit.eq.0.or.npra.eq.0)go to 6400
c     write(nlp,3040)lco,kro,pivot

      if(abs(pivot).lt.1d-28)go to 6410

c6400 continue

      r1(kro) = r1(kro)/pivot
      write(nd1) kro,lcol,lpivco,lhed(1),qq(1)
c
c     start back-substitution
c     ======================
      call bacsub
     1    (ntov ,ncod ,bc    ,r1    ,vel   ,press, maxfr,qq    ,lhed ,nd1,
     2     icho)
c
c     main exit with solution
c     ======================
6410  continue
c
3010  format(/'  nerror=',i5//
     1 '  the difference nmax-ncrit is not sufficiently large'
     1/'  to permit the assembly of the next element---'
     1/'  either increase nmax or lower ncrit'
     1/)
```

```
3030  format(/'  nerror=',i5//
     1 '  there are no more rows fully summed,this may be due to---'
     1/'  (1)incorrect coding of nop or nk arrays'
     1/'  (2)incorrect value of ncrit. increase ncrit to permit'
     1/'     whole front to be assembled'
     1/)

c3040  format(13h pivotal row=,i4,16h pivotal column=,i4,7h pivot=,e20.10
c    1)

3050  format('  warning-matrix singular or ill conditioned')

      return
      end
c
c     ================================================================
      subroutine gaussp(ngaus,xg,cg)

      implicit double precision(a-h,o-z)
c
c     x(g)   specifies the coordinates of the Gauss points
c     c(g)   specifies the Gauss weights
c
      dimension xg(3),cg(3)

      if(ngaus.eq.1) then
      xg(1)=0.0
      cg(1)=2.0
      elseif(ngaus.eq.2)  then
      xg(1) = 0.57735026919d00
      xg(2) = -xg(1)
      cg(1) = 1.00
      cg(2) = 1.00
      else
      xg(1) = 0.77459666924d00
      xg(2) = 0.0
      xg(3) = -xg(1)
      cg(1) = 0.55555555556d00
      cg(2) = 0.88888888889d00
      cg(3) = cg(1)

      endif
      return
      end

c     ================================================================
      subroutine getbcd (nbc  ,ibc  ,jbc  ,vbc
     1                  ,idv1 ,idv2 ,maxbc)
c
      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     nbc    number of nodal constraint data
c     ibc    array for constrained nodal points
c     jbc    array for constrained degree of freedom
c     vbc    array for boundary values
c     udv1   input device id.
c     idv2   output device id.
c     maxbc  see below
c
      dimension ibc  (maxbc) ,jbc  (maxbc),vbc  (maxbc)

      if (.not. eof(51))read (idv1,1010) (ibc(ind) ,jbc(ind) ,vbc(ind)
     1    ,ind=1,nbc)
      print*, "boundary conditions array read"
      write(idv2,3010)
      write(idv2,3020) (ibc(ind) ,jbc(ind) ,vbc(ind) ,ind=1,nbc)

      return

1010  format(2i5,f10.4)
3010  format(' ',//  /,' ',20('*'),' nodal constraint ',20('*'),//
     1' ',(8x,'id.',7x,'dof',10x,'value',10x)/)
3020  format(5x,i5,5x,i5,f17.4)

      end

c     ================================================================c
      subroutine getelm (nel  ,ncn  ,node ,idv1 ,idv2 ,maxel)
```

```
      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     ncn    number of nodes per element
c     node   array for element connectivity data
c     idv1   input device id.
c     idv2   output device id.
c     maxel  see below
c
      dimension node (maxel, ncn)

               do 6010 iel = 1 ,nel
6010  if (.not. eof(51))read (idv1,1010) iel ,(node(iel,icn),icn=1,ncn)
      print*, "nodal connectivity array read"
      write(idv2,3010)
                do 6020 jel = 1 ,nel
6020  write(idv2,3020) jel ,(node(jel,icn),icn=1,ncn)

      return

1010  format(21i7)
3010  format(' ',///,' ',20('*'),' element connectivity ',20('*'),//
     1' ',4x,'id.',7x,'n o d a l - p o i n t  e n t r i e s',/)
3020  format(21i7)

      end

c     ================================================================c
      subroutine getmat (nel ,nmat,pmat, idv1, idv2,maxel,rtem, rpef)
c
      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     nmat  number of materials
c     pmat  array for material constants for each element
c     idv1 input device id.
c     idv2  output devide id.
c     maxel see below
c
c     roden    density
c     rvisc    mu nought; consistency coefficient
c     pref     reference pressure
c     power    power law index
c     tref     reference temperature
c     tbco     coefficient b in the power law model
c     taco     coefficient a in the power law model
c     dispc    dispersion coefficient
c     gamad    shear rate

      dimension pmat (maxel,   9)

      write(idv2,3010)

      do 6010 imat = 1 ,nmat
      if (.NOT. EOF(51)) read(idv1,1010) rvisc, power, tref, tbco, taco,
     1                                   dispc, pref, roden, gamad
      print*, "material properties read"
                ifrom = 1
                ito   = nel

                if(rtem .eq.0.) rtem  = 0.001
                if(rpef .eq.0.) rpef  = 0.001

      do 6020 iel     = ifrom ,ito
         pmat(iel,1) = rvisc
         pmat(iel,2) = pref
         pmat(iel,3) = power
         pmat(iel,4) = tref
         pmat(iel,5) = tbco
         pmat(iel,6) = taco
         pmat(iel,7) = dispc
         pmat(iel,8) = roden
         pmat(iel,9) = gamad
         rtem        = tref
         rpef        = pref

6020  continue
```

```
      write(idv2,3030)
      write(idv2,3040) tref ,tbco, pref, taco
      write(idv2,3050)
      write(idv2,3060) dispc , roden , gamad

6010  continue

      return

1010  format(9d10.5)

3010  format(' ',//' ',35('*'),' material properties ',35('*'),//
     1      ' ',2x,'id.',5x,'eid.(from-to)',3x,'consistency co-efficient'
     2,5x,'power law index',/)
3020  format(' ',i3,i12,i4,5x,g15.5,15x,g15.5)
3030  format(/x,' reference temperature   coefficient b
     1   reference pressure  coefficient a '/)
3040  format(f16.3,f22.4,6x,g10.3,9x,g10.3)
3050  format(/x,
     1'Dispersion Coefficient     Density             Shear rate'/)
3060  format(g13.3,15x,g7.1,6x,g16.5)

      end

c    ===============================================================
      subroutine getnod (nnp  ,cord ,idv1 ,idv2 ,maxnp,ndim,icord)
c
      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     nnp    total number of nodal points in the mesh
c     cord   array for nodal coordinates
c     idv1   input device id.
c     idv2   output device id.
c     ndim   see below
c
      dimension cord(maxnp, ndim)
c
      if (.NOT. EOF(51)) read (idv1,1010) (jnp ,(cord(jnp,idf),idf=1,3)
     1    ,jnp=1,nnp)
      if(icord.eq.0) write(idv2,3010)
      if(icord.eq.1) write(idv2,3020)
      write(idv2,3030) (jnp ,(cord(jnp,idf),idf=1,3) ,jnp=1,nnp)

      return

1010  format(i7,e20.12,e20.12,e20.12)
3010  format(' ',///' ',20('*'),' nodal coordinates ',20('*'),//
     1' ',(7x,'id.',13x,'x-coord',13x,'y-coord',20x)/)
     1' ',(7x,'id.',13x,'x-coord',13x,'y-coord',13x,'z-coord',13x)/)
3020  format(' ',///' ',20('*'),' nodal coordinates ',20('*'),//
     1' ',2(7x,'id/',7x,'r-coord',7x,'z-coord',20x)/)
3030  format(' ',i10,10x,f10.6,10x,f10.6,10x,f10.6)

      end

c    ================================================================
      subroutine minimax
     1( cmax  , pmax  , vel  , press ,  maxnp,  nnp  ,    nc,
     2  np    , nm    , ncm  , nvxm  ,  nvym  ,  nvzm ,
     3  nvxl  , nvyl  , nvzl , pmin  ,  cmin  ,
     4  vxmax , vxmin , vymax, vymin ,  vzmax ,  vzmin,  ndim , maxdf )

      implicit real*8 (a-h,o-z)
      dimension vel  (maxdf)
      dimension press ( maxnp )

c
      vxmax   =   vel(1)
      vxmin   =   vel(1)
      vymax   =   vel(nnp+1)
      vymin   =   vel(nnp+1)
      vzmax   =   vel(2*nnp+1)
      vzmin   =   vel(2*nnp+1)

      pmax    =   press (1)
      pmin    =   press (1)

      nc      =   1
```

```
      np      =   1
      nm      =   1
      nvxm    =   1
      nvym    =   1
      nvzm    =   1
      nvxl    =   1
      nvyl    =   1
      nvzl    =   1

      do 6020 i=2,nnp

      pm      =   press (i)
      pi      =   press (i)
      vxmx    =   vel(i)
      vxmn    =   vel(i)
      vymx    =   vel(nnp+i)
      vymn    =   vel(nnp+i)
      vzmx    =   vel(2*nnp+i)
      vzmn    =   vel(2*nnp+i)

         if (   pm.gt.pmax  ) then
            pmax=pm
            np  =i
         endif
         if (   pi.lt.pmin  ) then
            pmin = pi
            nm   = i
         endif
         if ( vxmx.gt.vxmax ) then
            vxmax= vxmx
            nvxm  = i
         endif
         if ( vymx.gt.vymax ) then
            vymax= vymx
            nvym  = i
         endif
         if ( vzmx.gt.vzmax ) then
            vzmax= vzmx
            nvzm  = i
         endif

         if ( vxmn.lt.vxmin ) then
            vxmin= vxmn
            nvxl  = i
         endif
         if ( vymn.lt.vymin ) then
            vymin= vymn
            nvyl  = i
         endif
         if ( vzmn.lt.vzmin ) then
            vzmin= vzmn
            nvzl  = i
         endif
 6020 continue

      return
      end

c     ====================================================================
      subroutine output
     1     (nnp  ,vel  ,press, maxdf,maxnp,icord, stres)
c
      implicit double precision(a-h,o-z)
c
c     arguments are already defined
c     ===========================
      dimension vel(maxdf), press(maxnp)
      dimension stres(maxnp,    6)

      write(60,3010)

      if(icord.eq.0) write(60,3020)
      if(icord.eq.1) write(60,3030)

      do 6010 inp = 1,nnp
                         jnp = inp + nnp
                         knp = inp + (2*nnp)
      press(inp)=press(inp)
      write(60,3040)inp,vel(inp),vel(jnp),vel(knp),press(inp),
     1stres(inp,1)
```

```
       call minimax
      1( cmax   , pmax   , vel   , press ,  maxnp,  nnp  ,    nc,
      2  np     , nm     , ncm  , nvxm   ,  nvym   ,  nvzm ,
      3  nvxl   , nvyl   , nvzl , pmin   ,  cmin   ,
      4  vxmax , vxmin  , vymax, vymin  ,  vzmax ,  vzmin,  ndim , maxdf )

       write(60,3045)
       write(60,3050)nvxm,vel(nvxm),nvxl,vel(nvxl)

       write(60,3055)
       write(60,3060)nvym,vel(nnp+nvym),nvyl,vel(nnp+nvyl)

       write(60,3065)
       write(60,3070)nvzm,vel(2*nnp+nvzm),nvzl,vel(2*nnp+nvzl)

       write(60,3075)
       write(60,3080)np,press(np),nm,press(nm)


 3010  format(/' nodal velocities and  pressures  '/)
 3020  format('  id.      ux          uy            uz     press  stress'/)
 3030  format('  id.      ur          uz            uz             press'/)
 3040  format(i5,3e13.4,e22.8,g15.5)

 3045  format('node no.         max ux    node no.        min ux')
 3050  format(i5,e22.8,i5,e22.8,/)

 3055  format('node no.         max uy    node no.        min uy')
 3060  format(i5,e22.8,i5,e22.8,/)

 3065  format('node no.         max uz    node no.        min uz')
 3070  format(i5,e22.8,i5,e22.8,/)

 3075  format('node no.         max p     node no.        min p')
 3080  format(i5,e22.8,i5,e22.8,/)

       return
       end

 c     ==================================================================
       subroutine putbcv
      1 (nnp  ,nbc ,ibc ,jbc ,vbc ,ncod ,bc ,maxbc,maxdf,maxel,maxst,
      2  node)
 c
       implicit double precision(a-h,o-z)
 c
 c     arguments
 c     =========
 c     ncod    array for constraint switch defined for every d.o.f.
 c     bc      array for storing contraint value
 c     maxbc   see below
 c     maxdf   see below
 c
       dimension ibc  (maxbc) ,jbc  (maxbc) ,vbc  (maxbc)
       dimension ncod (maxdf) ,bc   (maxdf) ,node (maxel,maxst)

       do 6010 ind = 1 ,nbc
       if(jbc(ind)>4) goto 6010
       jnd          = ibc(ind)+(jbc(ind)-1)*nnp
       bc    (jnd) = vbc(ind)
       ncod  (jnd) =  1
 6010  continue

 c     specifying the stress free condition on node number 84
 c     ====================================================
 c     iel=16
 c     inp=24
 c     kc=iabs(node(iel,inp))
 c
       return
       end

 c     ==================================================================
       subroutine secinv
      1   (nel  ,nnp  ,ncn  ,ngaus,node ,sinv ,cord ,p  ,b ,
      2    del  ,da   ,vel  ,maxnp,maxel,maxst,ndim ,icord,
      3    maxdf,num)
 c
       implicit double precision(a-h,o-z)
```

```
c function
c --------
c     calculates the second invariant of rate of deformation
c     tensor at integration points.
c
      dimension vel  (maxnp, ndim) ,cord (maxnp, ndim)
      dimension node (maxel,maxst) ,sinv (maxel,   27)
      dimension p    (    20     ) ,del  (    3,   20)
      dimension b    (    3,   20)

                                               rewind 15
      do 5000 iel= 1 , nel
              lg = 0
      do 5010 ig = 1 ,ngaus
      do 5010 jg = 1 ,ngaus
      do 5010 kg = 1 ,ngaus

              lg = lg+1

           read (15) iiel,iig,jjg,p,del,b,da

                  u11 = 0.0
                  u12 = 0.0
                  u13 = 0.0
                  u21 = 0.0
                  u22 = 0.0
                  u23 = 0.0
                  u31 = 0.0
                  u32 = 0.0
                  u33 = 0.0

          do 5020 icn = 1 ,ncn
                                         jcn = iabs(node(iel,icn))

c *** components of the rate of deformation tensor

                  u11 = u11 + b(1,icn)*vel(jcn,1)
                  u12 = u12 + b(2,icn)*vel(jcn,1)
                  u13 = u13 + b(3,icn)*vel(jcn,1)
                  u21 = u21 + b(1,icn)*vel(jcn,2)
                  u22 = u22 + b(2,icn)*vel(jcn,2)
                  u23 = u23 + b(3,icn)*vel(jcn,2)
                  u31 = u31 + b(1,icn)*vel(jcn,3)
                  u32 = u32 + b(2,icn)*vel(jcn,3)
                  u33 = u33 + b(3,icn)*vel(jcn,3)

 5020     continue
c
c *** second invariant of the rate of deformation tensor
c
      sinv(iel,lg)=0.125*((u11+u11)*(u11+u11)+
     1                    (u12+u21)*(u12+u21)+
     2                    (u13+u31)*(u13+u31)+
     3                    (u21+u12)*(u21+u12)+
     4                    (u22+u22)*(u22+u22)+
     5                    (u23+u32)*(u23+u32)+
     6                    (u31+u13)*(u31+u13)+
     7                    (u32+u23)*(u32+u23)+
     8                    (u33+u33)*(u33+u33))

 5010 continue
 5000 continue

      return
      end

c     ==================================================================

      subroutine setprm
     1     (nnp  ,nel  ,ncn  ,node ,ndf  ,maxel,maxst,ndn  ,ntrix,
     2      maxdf,ntov ,mdf  ,nopp )
c
      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     all arguments are defined elsewhere.
c
      dimension node (maxel,maxst), ndn  (maxdf)
      dimension mdf  (maxdf      ), nopp (maxdf)
```

```
c      function
c      ========
c      Sets the location data for nodal degrees of freedom
c
       do 6010 iel = 1 ,nel
          ndn(iel) = ntrix
       do 6010 icn = 1 ,ncn
               kcn = node(iel,icn)
               jacn= icn+(ndf-3)*ncn
               lacn= kcn+(ndf-3)*nnp

               jbcn= icn+(ndf-2)*ncn
               lbcn= kcn+(ndf-2)*nnp

               jccn= icn+(ndf-1)*ncn
               lccn= kcn+(ndf-1)*nnp

       node(iel,jacn) = lacn
       node(iel,jbcn) = lbcn
       node(iel,jccn) = lccn

6010   continue
       do 6020 idf = 1,ntov
          mdf(idf) = 1
         nopp(idf) = idf
6020   continue

       return
       end
c
c      ==================================================================
       subroutine shape ( xi , eta , zeta, p ,del , ncn )
       implicit double precision (a-h,o-z)
c
       DIMENSION p(20) ,del(3,20)
       if (ncn.eq.8) then
          del(1,1)=-0.125*(1-eta)*(1-zeta)
          del(1,2)=-0.125*(1+eta)*(1-zeta)
          del(1,3)= 0.125*(1+eta)*(1-zeta)
          del(1,4)= 0.125*(1-eta)*(1-zeta)
          del(1,5)=-0.125*(1-eta)*(1+zeta)
          del(1,6)=-0.125*(1+eta)*(1+zeta)
          del(1,7)= 0.125*(1+eta)*(1+zeta)
          del(1,8)= 0.125*(1-eta)*(1+zeta)
c......................................................
          del(2,1)=-0.125*(1-xi)*(1-zeta)
          del(2,2)= 0.125*(1-xi)*(1-zeta)
          del(2,3)= 0.125*(1+xi)*(1-zeta)
          del(2,4)=-0.125*(1+xi)*(1-zeta)
          del(2,5)=-0.125*(1-xi)*(1+zeta)
          del(2,6)= 0.125*(1-xi)*(1+zeta)
          del(2,7)= 0.125*(1+xi)*(1+zeta)
          del(2,8)=-0.125*(1+xi)*(1+zeta)
c..........................................................
          del(3,1)=-0.125*(1-xi)*(1-eta)
          del(3,2)=-0.125*(1-xi)*(1+eta)
          del(3,3)=-0.125*(1+xi)*(1+eta)
          del(3,4)=-0.125*(1+xi)*(1-eta)
          del(3,5)= 0.125*(1-xi)*(1-eta)
          del(3,6)= 0.125*(1-xi)*(1+eta)
          del(3,7)= 0.125*(1+xi)*(1+eta)
          del(3,8)= 0.125*(1+xi)*(1-eta)
c..........................................................
          p(1)=0.125*(1-xi)*(1-eta)*(1-zeta)
          p(2)=0.125*(1-xi)*(1+eta)*(1-zeta)
          p(3)=0.125*(1+xi)*(1+eta)*(1-zeta)
          p(4)=0.125*(1+xi)*(1-eta)*(1-zeta)
          p(5)=0.125*(1-xi)*(1-eta)*(1+zeta)
          p(6)=0.125*(1-xi)*(1+eta)*(1+zeta)
          p(7)=0.125*(1+xi)*(1+eta)*(1+zeta)
          p(8)=0.125*(1+xi)*(1-eta)*(1+zeta)
c..........................................................

       endif
       return
       end

c      =================================================================
       subroutine stress
     1     (nel,nnp,ncn ,node ,p , b , da ,vel  ,maxnp, maxel,  maxst ,
```

```
     [illegible], stres, press, rvisc ,clump ,ngaus   ,

        implicit double precision(a-h,o-z)
c
c      function
c      --------
c      calculates stress components at integration points,
c      Direct Approach (coupled scheme)
c
       dimension node (maxel,maxst)  ,  b    (  3,  20)
       dimension stres(maxnp,    6)  ,  press(maxnp   ) , p (  20    )
       dimension vel  (maxnp,    3)
       dimension clump(maxnp      )

                                                             rewind 15

                                                  do 4990 inp   =1,maxnp
                                                  do 4990 icp   =1, 6
                                                  stres(inp,icp)= 0.0
 4990                                             continue


       do 5000 iel = 1 ,nel

       do 6010 ig=1,ngaus

       do 6010 jg=1,ngaus

       do 6010 kg=1,ngaus

        if(.not. EOF(15))read(15) iiel,iig,jjg,kkg,p ,del ,b , da

                  u11 = 0.0
                  u12 = 0.0
                  u13 = 0.0
                  u21 = 0.0
                  u22 = 0.0
                  u23 = 0.0
                  u31 = 0.0
                  u32 = 0.0
                  u33 = 0.0
                presl = 0.0

         do 6020 icn = 1 ,ncn
                                   jcn = iabs(node(iel,icn))
                  u11 = u11 + b(1,icn)*vel(jcn,1)
                  u12 = u12 + b(2,icn)*vel(jcn,1)
                  u13 = u13 + b(3,icn)*vel(jcn,1)
                  u21 = u21 + b(1,icn)*vel(jcn,2)
                  u22 = u22 + b(2,icn)*vel(jcn,2)
                  u23 = u23 + b(3,icn)*vel(jcn,2)
                  u31 = u31 + b(1,icn)*vel(jcn,3)
                  u32 = u32 + b(2,icn)*vel(jcn,3)
                  u33 = u33 + b(3,icn)*vel(jcn,3)
                presl = presl + p(icn)*press(jcn)

 6020    continue

c
c     cartesian components of the stress tensor
c     =========================================
c     Shear Stress (Tau)
c     =========================================

       sd11 = 2.0    *rvisc *  u11
       sd22 = 2.0    *rvisc *  u22
       sd33 = 2.0    *rvisc *  u33
       sd12 = rvisc * (u12 +  u21)
       sd13 = rvisc * (u13 +  u31)
       sd23 = rvisc * (u23 +  u32)

c     =========================================
c     Normal Stress (Pi)
c     =========================================

       s11  =-presl + sd11
       s22  =-presl + sd22
       s33  =-presl + sd33
       s12  = sd12
       s13  = sd13
```

```
c    ================================================================
c *** calculate stress at nodal points
c    ================================================================

         do 6500 icn = 1 ,ncn
                                    jcn = iabs(node(iel,icn))

                     stres(jcn,1)= stres(jcn,1)
     1                           + p(icn) *s11  *da
                     stres(jcn,2)= stres(jcn,2)
     1                           + p(icn) *s22  *da
                     stres(jcn,3)= stres(jcn,3)
     1                           + p(icn) *s33  *da
                     stres(jcn,4)= stres(jcn,4)
     1                           + p(icn) *s12  *da
                     stres(jcn,5)= stres(jcn,5)
     1                           + p(icn) *s13  *da
                     stres(jcn,6)= stres(jcn,6)
     1                           + p(icn) *s23  *da

 6500    continue

 6010 continue

 5000 continue

      return

      end

c     ==================================================================
      subroutine visca
     1          (rvisc,power,visc,stemp,rtem,tbco,spress,rpef,taco
     2          ,gamad )

      implicit double precision(a-h,o-z)

        visc   = rvisc*(4.0*gamad**((power-1.0)*0.5))
     1           *exp(-tbco*(stemp-rtem))

      return
      end
c
c     ==================================================================
c     e    n    d    o    f    p    r    o    g    r    a    m
c     ==================================================================
```

3DFEANOF

– P version

3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow – Continuous Penalty version

Fortran Program Source Code

1. Main Program Code
2. Subroutines

Kee Chien Ting

Advanced Separation Techniques Group
Department of Chemical Engineering
Loughborough University

```
c
c     This is a program for the solution of non-newtonian, isothermal,
c     incompressible flow problems using the weighted residual galerkin
c     finite element method
c
c     The solution scheme is based on the Penalty Scheme
c
c     Velocity components are the prime unknowns in the flow
c     field.
c
c     This program uses eight-noded linear prism element.
c
c     Algebraic equations are solved by a frontal method.
c
c     A complete list of options is given on the program listing.
c
c     The program consists of a main module and subroutines
c
c     The program is written in FORTRAN programming language
c
c     This program is developed by Kee Chien Ting (last revised Nov 2004)
c
c     ================================================================
c
c     work files
c     ==========
c       unit      contents
c ====================================================================
c         51   i    input data file
c              i
c         60   i    output file for documentation
c              i
c         11   i    output file containing velocity field data for
c              i    plotting
c              i
c         14   i    used as a work file in the solver routine
c              i
c         15   i    stores shape functions and their derivatives at
c              i    'full' integration points
c              i
c         17   i    output file containing pressure data for
c              i    contour plotting
c              i
c         20   i    output file containing elemental stiffness matrix
c              i    for element number 14 as seen on the mesh
c              i
c ====================================================================
c
c       List of variables
c       =================
c       aa   (   27,  27)   element coefficient matrices on LHS
c       K    (   27,  27)   element coefficient matrices on RHS
c       b    (    3,  20)   global derivatives of shape functions
c       bc   (maxdf     )   nodal constraints (boundary conditions)
c       cord (maxnp,ndim)   nodal coordinates
c       del  (    3,  20)   local  derivatives of shape functions
c       vel  (maxdf     )   nodal velocities  (displacements)
c       dsc1, dsc2          depths of slip layers
c       grav1               first  component of the applied body force
c       grav2               second component of the applied body force
c       icord               indicates whether the coordinate system is cartesian (planar) or
c                           cylindrical (axisymmetric)
c       tolp                convergence tolerance factor for pressures
c       tolv                convergence tolerance factor for velocities
c       nbc                 total number of boundary-node constraints
c       ncn                 number of nodes per element
c       ndf                 degree of freedom per node
c       ndim                dimensions of the solution domain
c       nel                 total number of elements
c       ngaus               number of integration points
c       nnp                 total number of nodal points
c       node (maxel,maxst)   element connectivity
c       nter                maximum number of iterations for non-newtonian case
c       num                 number of integration points per element
c       p    (    20    )   shape functions
c       press(maxnp     )   nodal pressures
c       r1   (maxdf     )   global load vector  (r.h.s.)
c       rfrct               friction coefficient (slip)
c       rr   (    27    )   element load vector
```

```
c     [illegible]                  global stiffness matrix ( a in ax=r.h.s.)
c     rvisc                   mu nought;consistency coefficient in power-law model
c     power                   power law index
c     stemp                   temperature
c     rtem                    reference temperature
c     spress                  pressure
c     rpress                  reference pressure
c     tco                     coefficient relating viscosity to temperature
c     pco                     coefficient relating viscosity to pressure
c     gamad                   shear rate
c     nwr                     no. of sample nodes for recording transient solutions
c
c     ================================================================
c     List of Subroutines
c     ===================
c
c     bacsub                  backsubstitution method for finding the final
c                             solution vector
c     clean                   cleans the arrays and prepares them for
c                             solution
c     contol                  makes a check for the convergence
c     deriv                   calculates the jacobian matrix, its determinant
c                             and global derivatives of the shape functions
c     flow                    calculates the velocities and pressures
c     front                   frontal method for solving the final set
c                             of equations
c     gaussp                  specifies the gauss points and weights for
c                             quadrature integration
c     getbcd                  specifies the primary boundary conditions
c     getelm                  specifies the nodal connectivity array
c     getmat                  reads the input material data
c     getnod                  reads the nodal co-ordinates for cartesian
c                             and axisymmetric systems
c     lumpm                   evaluates the terms of the mass matrix
c     minimax                 captures the minimum and maximum value for
c                             each specified variables
c     output                  prints the final solution
c     putbcv                  imposes the primary boundary conditions for
c                             velocity
c     secinv                  calculates the second invariants
c     setprm                  Sets the location data for nodal degrees of
c                             freedom
c     shape                   calculates the shape functions and their
c                             derivatives
c     stress                  calculates stress components at integration
c                             points
c     visca                   calculates the viscosity
c
c ================================================================
      parameter (maxel  = 30000    )
      parameter (maxnp  = 37000    )
      parameter (maxbc  = 20000    )
      parameter (maxdf  = maxnp*3  )
      parameter (maxst  = 60       )
      parameter (maxfr  = 2000     )
      parameter (ndim   = 3        )

      implicit real*8 (a-h,o-z)
c
c     Storage allocation
c     ==================
c
      dimension title (          80)
      dimension node  (maxel,maxst) ,pmat (maxel, 10) ,cord (maxnp,ndim)
      dimension ncod  (maxdf       ), bc   (maxdf     )
      dimension ibc   (maxbc       ) ,jbc  (maxbc     ) ,vbc  (maxbc     )
      dimension vel   (maxdf       ) ,press(maxnp     )
      dimension r1    (maxdf       )
      dimension clump (maxnp       ) ,stres(maxnp, 11)
      dimension vet   (maxdf       ) ,cet  (maxnp     ) ,pet  (maxnp     )
      dimension nopp  (maxdf       )
      dimension aa    (maxst,maxst) ,rr   (maxst     )
      dimension xg    (           3) ,cg   (          3)
      dimension p     (          20) ,del  (    3, 20) ,b    (    3,  20)
      dimension ldest (maxst       ) ,kdest(maxst     ) ,nk   (maxst     )
      dimension eq    (maxfr,maxfr) ,lhed (maxfr     ) ,khed (maxfr     )
      dimension kpiv  (maxfr       ) ,lpiv (maxfr     ) ,jmod (maxfr     )
      dimension qq    (maxfr       ) ,pvkol(maxfr     ) ,sinv (maxel, 35)
      dimension mdf   (maxdf       ) ,ndn  (maxdf     )
      dimension ldsc  (22          )
      dimension temp  (maxnp       )
      character *20 filnam
c
c    Opening of input and output data files
c    ======================================
c
      print*,'enter the name of your data file'
      read(*,2000) filnam

      open(unit=51,File=filnam,access='sequential',form='formatted',
     1     status="unknown",iostat=ios)

      open(unit=60,file='res.txt',access='sequential',form='formatted',
     1     status='unknown",iostat=ios)

      open(unit=17,file='stress.txt',access='sequential',form='formatted',
     1     status="unknown",iostat=ios)

      open(unit=14,form='unformatted',status='scratch',iostat=ios)
      open(unit=15,form='unformatted',status='scratch',iostat=ios)
      open(unit=16,form='unformatted',status='scratch',iostat=ios)

        if(ios==0)then
              print*,"files opened"
        else
              print*,"files not opened"
        stop
        end if

      rewind 51
      rewind 60
      rewind 20
c
c     Initialize arrays
c    =================

      do 5010 itl = 1,maxel
      do 5010 ivl = 1,maxst
                          node (itl,ivl) = 0
5010  continue
      do 5020 itl = 1,maxel
      do 5020 ivl = 1,10
                          pmat (itl,ivl) = 0.0
5020  continue
      do 5030 itl = 1,maxnp
      do 5030 ivl = 1,3
                          cord (itl,ivl) = 0.0
5030  continue
      do 5040 itl = 1,maxnp
      do 5040 ivl = 1,11
                          stres(itl,ivl) = 0.0
5040  continue
      do 5050 itl = 1,maxdf
                          vel  (itl    ) = 0.0
5050  continue
      do 5060 itl = 1,maxdf
                          ncod   (itl)   = 0
                          rl     (itl)   = 0.0
                          bc     (itl)   = 0.0
                          vet    (itl)   = 0.0
                          mdf    (itl)   = 0
                          ndn    (itl)   = 0
                          nopp   (itl)   = 0
5060  continue
      do 5070 itl = 1,maxnp
                          clump  (itl)   = 0.0
                          cet    (itl)   = 0.0
                          pet    (itl)   = 0.0
                          press  (itl)   = 0.0
5070  continue
      do 5080 itl = 1,maxbc
                          ibc   (itl)    = 0
                          jbc   (itl)    = 0
                          vbc   (itl)    = 0.0
5080  continue
      do 5090 itl = 1,20
                          del   (1,itl)  = 0.0
                          del   (2,itl)  = 0.0
                          del   (3,itl)  = 0.0
5090  continue
      do 5100 itl = 1,maxst
```

```
                          idest (itl  )  = 0
                          kdest (itl  )  = 0
                          nk    (itl  )  = 0
5100  continue
      do 5110 itl = 1,maxfr
                          lhed  (itl  )  = 0
                          khed  (itl  )  = 0
                          kpiv  (itl  )  = 0
                          lpiv  (itl  )  = 0
                          jmod  (itl  )  = 0
                          qq    (itl  )  = 0.0
                          pvkol (itl  )  = 0.0
      do 5110 ill = 1,maxfr
                          eq    (itl,ill)= 0.0
5110  continue
      do 5120 itl = 1,maxel
      do 5120 ivl = 1,35
                          sinv  (itl,ivl)  = 0.0
                          rmat1  (itl,ivl) = 0.0
                          rmat2  (itl,ivl) = 0.0
5120  continue

c---------------------------------------------------------------------------

c     Title of the program
c     ====================

      if(.not. eof(51)) read (51,2010) title
      write(60,4010) title

c     Element description data
c     ========================

      if (.not. eof(51)) read (51,2020) ncn ,ngaus
      print*, "ncn, ngaus read"
      write(60,4020) ncn ,ngaus

c     Mesh, boundary condition and material parameters
c     ================================================

      if (.not. eof(51)) read (51,2030) nnp  ,nel  ,nbc  ,nmat
      print*, "nnp, nel ,nbc ,nmat read"
      if (.not. eof(51)) read (51,2040) ntep ,icord
c
      if(icord.eq.0) write(60,4030)
      if(icord.eq.1) write(60,4040)

      write(60,4050)

      if(ntep.eq.0) ntep=1

c    ============================================================
c    icord=0 coordinate system is cartesian   (   planar   )
c    icord=1 coordinate system is cylindrical (axisymmetric)
c
c    if ntep = 1 then computed result after every iteration will
c    be printed ;if you do not need the result of intermediate
c    computations choose your own ntep;the result of first and
c    converged solutions will always be printed.
c    ============================================================

                if(nnp   .eq.0  .or.nnp  .gt.maxnp) then
                                             write(60,4060)
            elseif(nel   .eq.0  .or.nel  .gt.maxel) then
                                             write(60,4060)
            elseif(nbc   .eq.0  .or.nbc  .gt.maxbc) then
                                             write(60,4060)
            elseif(nmat  .eq.0  .or.nmat .gt.maxel) then
                                             write(60,4060)

            print*, "the program is aborted"
            stop

                endif

      write(60,4070) nnp ,nel ,nbc ,nmat

      if (.not. eof(51)) read (51,2050) grav1, grav2, grav3
      print*, "grav1 grav2 grav3 read"
      write(60,4080) grav1, grav2, grav3
      print*, "tolv, tolp, tolc read"

      maxer=maxel

c     ========================================================================
c     Read input data from main data file and prepare arrays for solution process
c     ========================================================================

       call getmat(nel,nmat,pmat,51,60,maxel,rtem,rpef)
       call getnod(nnp,cord,51,60,maxnp,ndim,icord)
       call getelm(nel,ncn,node,51,60,maxer)
       call getbcd(nbc,ibc,jbc,vbc,51,60,maxbc)


c     =======================================================================
c                              Start of the time loop
c     =======================================================================
c
c      Set control parameters (default values are overwritten by input data
c      if specified)
c
c      ncn      number of nodes per element
c      ngaus    number of integration points
c      nter     maximum number of iterations for non-newtonian case
c      ndim     number of space dimensions in the solution domain
c
c       nter  = 5

                                                do 5130 ivel= 1,maxdf
                                                vel (ivel)  =   0.0
5130                                            continue
                                                do 5140 item= 1,maxnp
                                                temp(item)  =    rtem
5140                                            continue

c     ===========================================================================
c                                     Transient data
c     ===========================================================================
c
c      stime       starting time
c
c      deltat      time increment
c
c      alpha       indicates the choice of method being employed in alpha
c                  time stepping technique (backward difference,
c                  forward difference, central difference, galerkin)
c
c      nter        maximum number of time steps being employed for finding solution
c     ==========================================================================

       print*,"  "
       print*,"Enter the number of time steps desired"
       read*, nter
       write(60,4100) nter

       print*,"Enter the delta t desired"
       read*, deltat

       write(60,4110) deltat

       print*,"Type Code for taylor galerkin technique being used"
       print*,"Code 1:Forward Diference Method"
       print*,"Code 2:Galerkin's Method"
       print*,"Code 3:Central Diference Method"
       print*,"Code 4:Backward Diference Method"
       print*,"Code 5:Temporal Upwinding"

       read*, Code

           if(code .eq. 1) then
           alpha=0.0
           else if(code .eq. 2) then
           alpha=0.5
           else if(code .eq. 3) then
           alpha=2.0/3.0
           else if(code .eq. 4) then
           alpha=1.0
           else if(code .eq. 5) then
           print*,"Type in the value of alpha between 0.5 & 1"
           read *, alpha
```

```
          ----
          print*, "Error in Typing code"
          end if

      print*, "alpha=",alpha

      do 5150 iter = 1 ,nter
      print*,'iter=',iter
      time = iter*deltat

      write(60,4090) iter

c     ==================================================================================
c                                 Calculate Nodal Velocities & Pressures
c     ==================================================================================
      icho=1
                                                     rewind 11
                                                     rewind 14
                                                     rewind 15
                                                     rewind 16

                                ndf   = 3
                                ntov  = ndf * nnp
                                ntrix = ndf * ncn
      call clean
     1    (ncn  ,nel  ,ndf  ,node ,r1   ,maxel,maxst,maxdf,
     2     bc   ,ncod ,icho )

      call setprm
     1    (nnp  ,nel  ,ncn  ,node ,ndf  ,maxel,maxst,ndn  ,ntrix,
     2     maxdf,ntov ,mdf  ,nopp )

      call putbcv
     1  (nnp ,nbc ,ibc ,jbc ,vbc ,ncod ,bc,maxbc,maxdf,maxel,maxst,
     2   node)


c     idv4 is the file specifier for unit=20
c     ===================================

      idv4=20

      do 5160 iel=1,nel

      call flow (node ,cord ,pmat ,nopp ,mdf ,ndn ,ncod ,bc ,vel
     1,press, r1, temp,ldest,kdest,nk ,eq ,lhed ,khed ,kpiv ,lpiv,
     2jmod, qq, pvkol,iter ,nel ,ncn , ngaus,grav1 ,grav2, grav3, p,
     3del, b, ntrix, maxel, maxnp, maxst, maxfr, maxdf, ndim ,
     4aa ,xg ,da ,ntov ,num, icord, rr, iel, del1,deltat, alpha,idv4,
     5sinv, icho, nnp, tref,rmat1,rmat2)

5160  end do

c     ===================================================================================
c                 calculates the second invariant of rate of deformation
c                            tensor at integration points.
c     ===================================================================================
      call secinv
     1   (nel  ,nnp  ,ncn  ,ngaus,node ,sinv ,cord ,p  ,b,
     2    del  ,da   ,vel  ,maxnp,maxel,maxst,ndim ,icord,
     3    maxdf,num)

c     =============================================================
c                                  Convergence check
c     =============================================================
      call contol(vel ,iter ,ntov ,nnp,maxnp,maxdf,errov, errop
     1,vet ,cet, pet, press)

c     =============================================================
c *** calculation of the nodal stress;using variational recovery
c     =============================================================
      call lumpm
     1     (clump,nnp  ,maxnp,nel  ,ngaus,p   ,del   ,b  ,maxst,
     2      node ,maxel,ncn  )

      call stress
     1      (nel,nnp,ncn ,node ,p , b , da ,vel  ,maxnp, maxel,  maxst ,
     2       maxdf, stres, press, rvisc ,clump ,ngaus,rmat1,rmat2  )

c     ================
c     Print the output
```

```
      iiter=(iter/ntep)*ntep

      if(iter.eq.1.or.iiter.eq.iter) then

      call output
     1   (nnp  ,vel  ,press, maxdf,maxnp,icord, stres)
      endif
c
c     ================================================================
c                          End of time loop
c     ================================================================

5150  continue

      close(51)
      close(unit=60)
      close(unit=11)
      close(unit=14)
      close(unit=15)
      close(20)
c
c    =================================================================
c                          Read statements
c    =================================================================
c
2000  format(a)
2010  format(80a)
2020  format(2i5)
2030  format(4i5)
2040  format(2i5)
2050  format(3f10.0)
2060  format(3f10.5)
c
c    =================================================================
c                          Write statements
c    =================================================================

4010  format(' ',5(/),' ',20x,60('*'),/' ',20x,'*',58x,'*',/
     1' ',20x,'*','  A  three  dimensional finite element model of a  ',
     29x,'*',/' ',20x,'*','  non-newtonian isothermal flow using ',
     320x,'*',/' ',20x,'*','  the UVP method.  ',39x,'*',/' ',20x,'*',
     558x,'*',/' ',20x,60('*')///,' ',20x,80('-'),/' ',20x,80a,/' ',
     620x,80('-'),///)

4020  format(' ',20x,3('['),' element description data',10('.'),/
     125x,'no.of nodes per element                      =',i10,/
     225x,'no.of integration points                     =',i10,/
     3//)

4030  format('    *** coordinate system is cartesian (planar) ***')
4040  format('*** coordinate system is cylindrical(axisymmetric) ***')
4050  format(' ')
4060  format(' ',10('['),'input data unacceptable',10(']')///)

4070  format(' ',20x,3('['),' mesh description data ',10('.'),/
     125x,'no.of nodal points                           =',i10,/
     225x,'no.of elements                               =',i10,/
     325x,'no.of nodal constraints on boundary          =',i10,/
     425x,'no.of different materials                    =',i10,//)

4080  format(' ',20x,3('['),' uniform body force vector ',10('.'),/
     125x,'grav1                                        =',f15.4,/
     225x,'grav2                                        =',f15.4,/
     325x,'grav3                                        =',f15.4,//)
4090  format(///'     iteration no.',i5,//)
4100  format(///' Total number of time steps  =',i5,//)
4110  format(///' Deltat                      =',f15.10,//)

c     ======================================================================
      end program
c     ======================================================================

c     ======================================================================
      subroutine bacsub
     1    (ntotl,ifix ,vfix ,rhs  ,soln ,mfrnt,rwork,iwork,idv2,
     2     icho )

      implicit double precision(a-h,o-z)
      dimension ifix (ntotl),vfix (ntotl),rhs (ntotl),soln (ntotl)
```

```
      dimension rwork(mfrnt) ,iwork(mfrnt) ,soln1(ntotl)

                        do 6010 ipos=1,ntotl
                        soln(ipos) =0.0
                        if(ifix(ipos).ne.0) soln(ipos)=vfix(ipos)
6010                    continue

      do 6020 kpos=1,ntotl

      backspace idv2
      read(idv2)     ipos,ifrnt,jfrnt,(iwork(k),rwork(k),k=1,ifrnt)
      backspace idv2

      if(ifix(ipos).ne.0)  go to 6020

      ww            = 0.0
      rwork(jfrnt) = 0.0

                        do 6030 k=1,ifrnt
                        jpos=iabs(iwork(k))
                        ww  =ww - rwork(k)*soln(jpos)
6030                    continue

      soln (ipos)=rhs(ipos)+ww

6020  continue

      if (icho .eq. 2) goto 6050

6050  continue

      return
      end
c
c    ===================================================================
c      subroutine clean
     1    (ncn  ,nel  ,ndf  ,node ,r1   ,maxel,maxst,maxdf,
     2     bc   ,ncod ,icho )

      implicit double precision(a-h,o-z)

c     arguments
c     =========
c     arguments are already defined
c
      dimension r1   (maxdf) ,node(maxel,maxst)
      dimension bc   (maxdf) ,ncod(maxdf      )

c     function
c     ========
c     cleans the used arrays and makes them ready for  solution
c
         do  6010 i   = 1,maxdf
                r1(i) = 0.0
                bc(i) = 0.0
              ncod(i) = 0
6010                continue
              ntrix   = ndf *ncn
         do  6020 iel = 1,nel
         do  6020 inp = 1,ntrix
         node(iel,inp) = iabs(node(iel,inp))
6020                continue
         if(icho.ne.1)then
         do 6030 iel = 1,nel
         write(11,3010)iel,(node(iel,j),j=1,ncn)
3010     format(10i5)
6030     continue

         endif

      return
      end
c
c    ===================================================================
      subroutine contol
     1(vel  ,iter ,ntov ,nnp  ,maxnp,maxdf,errov,errop,vet ,cet,
     2 pet, press)

      implicit double precision(a-h,o-z)

      dimension vel  (maxdf),press(maxnp)
      errv  = 0.0
      torv  = 0.0
      errp  = 0.0
      torp  = 0.0
c
c    calculate difference between velocities in consecutive iterations
c    ================================================================
c
                     do 6010 icheck = 1,ntov
      if(iter.eq.1) vet(icheck) = 0.0
                     errv = errv +
     1          (vel(icheck)-vet(icheck)) * (vel(icheck)-vet(icheck))
                     torv = torv + vel(icheck)*vel(icheck)

      vet(icheck) = vel(icheck)

6010                 continue
                     errov= errv/torv

c     calculate difference between pressures in consecutive iterations
c     ================================================================

                     do 6030 icheck = 1,nnp
      if(iter.eq.1)  pet(icheck) = 0.0
                          errp = errp +
     1       (press(icheck)-pet(icheck))*(press(icheck)-pet(icheck))
                     torp = torp + press(icheck)*press(icheck)

      pet(icheck) = press(icheck)

6030                 continue
                     errop= errp/torp
      return
      end

c     =================================================================
      subroutine deriv
     1    (iel  ,ig   ,jg   ,kg, p    ,del  ,b ,ncn ,da ,cg ,node,
     2     cord ,maxel,maxnp)

      implicit double precision(a-h,o-z)
      dimension p(20),b(3,40),del(3,20),cg(3),cj(3,3),cji(3,3)
      dimension node(maxel,27),cord(maxnp,3)

      do 6010 j=1,3
      do 6010 l=1,3
      gash=0.0
      do 6020 k=1,ncn
      nn=iabs(node(iel,k))

6020  gash=gash + del(j,k)*cord(nn,l)

      cj(j,l)=gash

6010  continue

      detj =  cj(1,1)*cj(2,2)*cj(3,3)+cj(2,1)*cj(3,2)*cj(1,3)
     1      + cj(1,2)*cj(2,3)*cj(3,1)-cj(1,3)*cj(2,2)*cj(3,1)
     2      - cj(1,2)*cj(2,1)*cj(3,3)-cj(2,3)*cj(3,2)*cj(1,1)

      if(detj.le.0.0) then
      write(60,3010) iel,detj
3010  format(1x ,' Error: Zero or Negative Jacobian. ', i6,g20.5)
      stop

      endif

      cji(1,1) =  (cj(2,2)*cj(3,3)-cj(3,2)*cj(2,3))  / detj
      cji(1,2) = ((cj(1,2)*cj(3,3)-cj(3,2)*cj(1,3))) / detj
      cji(1,3) =  (cj(1,2)*cj(2,3)-cj(2,2)*cj(1,3))  / detj
      cji(2,1) = ((cj(2,1)*cj(3,3)-cj(3,1)*cj(2,3))) / detj
      cji(2,2) =  (cj(1,1)*cj(3,3)-cj(3,1)*cj(1,3))  / detj
      cji(2,3) = ((cj(1,1)*cj(2,3)-cj(2,1)*cj(1,3))) / detj
      cji(3,1) =  (cj(2,1)*cj(3,2)-cj(3,1)*cj(2,2))  / detj
      cji(3,2) = ((cj(1,1)*cj(3,2)-cj(3,1)*cj(1,2))) / detj
      cji(3,3) =  (cj(1,1)*cj(2,2)-cj(2,1)*cj(1,2))  / detj

      endif
```

```
      do 6030 j=1,3
      do 6030 l=1,ncn
      b(j,l)=0.0
      do 6030 k=1,3
6030  b(j,l) = b(j,l) + cji(j,k) * del(k,l)

      da= detj*cg(ig)*cg(jg)*cg(kg)

      return
      end

c     ================================================================
      subroutine flow(node ,cord ,pmat ,nopp ,mdf ,ndn ,ncod ,bc ,vel
     1,press, r1, temp,ldest,kdest,nk ,eq ,lhed ,khed ,kpiv ,lpiv,
     2jmod, qq, pvkol, iter ,nel ,ncn , ngaus,grav1,
     3grav2, grav3, p, del, b, ntrix, maxel, maxnp, maxst, maxfr, maxdf,
     4ndim ,aa ,xg ,da ,ntov ,num, icord, rr, iel, del1,deltat,alpha,
     5idv4,sinv, icho, nnp, tref,rmat1,rmat2)

      implicit double precision(a-h,o-z)

      dimension node (maxel,maxst),pmat (maxel,  10),cord (maxnp, ndim)
      dimension ncod (maxdf      ),bc   (maxdf      ),sinv (maxel,   35)
      dimension vel  (maxnp ,   3),r1   (maxdf      )
      dimension aa   (maxst,maxst),rr   (maxst      ),ldest(maxst       )
      dimension xg   (           3),cg   (          3),kdest(maxst       )
      dimension x    (           3),v    (          3),nk   (maxst       )
      dimension bicn (           2),hh   (          3)
      dimension p    (          20),del  (   3,    20),b    (   3,     20)
      dimension eq   (maxfr,maxfr),nopp (maxdf      )
      dimension ldsc (          22)
      dimension lhed (maxfr      ),khed (maxfr      ),jmod (maxfr       )
      dimension lpiv (maxfr      ),kpiv (maxfr      ),qq   (maxfr       )
      dimension pvkol(maxfr      ),mdf  (maxdf      ),ndn  (maxdf       )
      dimension ppp  (20     ,  20),pp   (20         )
      dimension ak   (100,100)
      dimension akf  (100     )
      dimension NQ   (20     ,  20),NP   (3     ,    4)
      dimension C    (maxst      ),temp (maxnp      )
      dimension DEL1 (3          )
      dimension press(maxnp      ),clump(maxnp      ),SHAPE1D(3         )
      dimension gdsf (    3,   20)
      dimension dmass(100, 100)
      dimension rmat1 (maxel,   35) ,rmat2(maxel, 35)

                                            rvisc = pmat(iel,1)
                                            rpef  = pmat(iel,2)
                                            power = pmat(iel,3)
                                            rtem  = pmat(iel,4)
                                            tbco  = pmat(iel,5)
                                            taco  = pmat(iel,6)
                                            roden = pmat(iel,8)
                                            gamad = pmat(iel,9)
                                            rbulk = pmat(iel,10)

      velsound = 1150.0
      beta     =    0.0
      permx    = 1.0E-5
      permy    = 1.0E-5
      permz    = 1.0E-5
      Penal    = 10.0E+10
      mgaus    = ngaus - 1

                 do 6010 idf= 1,ntrix
                 rr (idf)      = 0.0
                 akf(idf)      = 0.0
                 C  (idf)      = 0.0
                 do 6010 jdf= 1,ntrix

                 aa   (idf,jdf)=0.0
                 dmass(idf,jdf)=0.0
                 ak   (idf,jdf)=0.0
6010             continue

c ------------------- parameters for penalty method --------------------
c
                     pm1 =1.0
                     pm2 =0.0
c
c     ----------------------------------------
c     ired = 1 for full integration
```

```
c      ---------------------------------------

       ired = 1

       if(ired.eq.1) then
c
c --- 'full' integration ---------------------------------------------
c
       call gaussp(ngaus,xg,cg,ncn)

              lg=0

       do 6020 ig=1,ngaus
                                    g = xg(ig)
       do 6020 jg=1,ngaus
                                    h = xg(jg)
       do 6020 kg=1,ngaus
                                    f = xg(kg)


                                   lg = lg + 1


       if(iter.eq.1) then


       call shape (g,h,f,p ,del , ncn, node, cord
      1                  nnp, maxel, maxst, maxnp, ndim)

       call deriv (iel,ig,jg,kg,p,del,b,ncn,da,cg,node,cord,
      1            maxel,maxnp)

                     iig=ig
                     jjg=jg
                     kkg=kg

                     write(15)   iel ,ig ,jg ,kg, p ,del ,b ,da

       else
       if(.not. EOF(15))read(15) iel,iig,jjg,kkg,p ,del ,b , da

       endif
c
c      calculation of viscosity based on the constitutive equation.
c
       spress = 0.0
       stemp  = 0.0

                 do 5333 ip = 1,ncn
                 jp     = iabs(node(iel,ip))
                 stemp = stemp + temp(jp) * p(ip)
 5333            continue
                 epsii = 1.d-10
                 gamad = sinv(iel,lg)
                 if(gamad.lt.epsii) gamad = epsii

       call visca
      1(rvisc,power,visc,stemp,rtem,tbco,spress,rpef,taco,gamad)


c -----calculate viscosity dependent penalty parameter--------------

       bulk = rbulk * visc

       rmat1(iel,lg) = visc

       rmat2(iel,lg) = bulk
c------------------------------------------------------------------
c      preparation of the convective acceleration terms/balancing
c      dissipation is used
c
                        do 6050 idff= 1,3
                        x(idff)     = 0.0
                        v(idff)     = 0.0
                        hh(idff)    = 0.0
6050                    continue
           do 6060 icn = 1 ,ncn
                   jcn = iabs(node(iel,icn))
           do 6060 idff= 1 ,  3
           x(idff)     = x(idff) + p(icn)*cord(jcn,idff)
```

```
              v(idff)         = v(idff) + p(icn)*vel (jcn,idff)
6060      continue


       if(icord.eq.1) then
c
c      modify da for axisymmetric computations.
c
       da = da * x(1)
       endif


c      column index

       do 6070 i=1,ncn

                      j11= i
                      j12= i + ncn
                      j13= i + 2*ncn
                      j14= i + 3*ncn

       do 6070 j=1,ncn
                      j21= j
                      j22= j + ncn
                      j23= j + 2*ncn
                      j24= j + 3*ncn


c      Dicretized form of 3D Stokes Equation
c
c --- Stiffness Matrix of Left Hand Side ('full' integration)--------------------
c
c      For Transient state (Cartesian co-ordinate system)


       aa(j11,j21)=aa(j11,j21) +  p(i)*p(j)*da
      2                             *pm1
      1                         +  alpha*deltat*((2.0*visc/roden)
      2                            +((1.0/roden)*bulk))
      3                            *b(1,i)*b(1,j)*da
      2                             *pm2
      3                         +  alpha*deltat*(visc/roden)*b(2,i)
      4                            *b(2,j)*da
      2                             *pm1
      5                         +  alpha*deltat*(visc/roden)*b(3,i)
      6                            *b(3,j)*da
      2                             *pm1

       aa(j11,j22)=aa(j11,j22) +  alpha*deltat*(visc/roden)
      1                            *b(2,i)*b(1,j)*da
      2                             *pm1
      2                         +  alpha*deltat*((1.0/roden)*bulk)
      3                            *b(1,i)*b(2,j)*da
      2                             *pm2

       aa(j11,j23)=aa(j11,j23) +  alpha*deltat*(visc/roden)*b(3,i)
      1                            *b(1,j)*da
      2                             *pm1
      2                         +  alpha*deltat*((1.0/roden)*bulk)
      3                            *b(1,i)*b(3,j)*da
      2                             *pm2


       aa(j12,j21)=aa(j12,j21) +  alpha*deltat*(visc/roden)*b(1,i)
      1                            *b(2,j)*da
      2                             *pm1
      2                         +  alpha*deltat*((1.0/roden)*bulk)
      3                            *b(2,i)*b(1,j)*da
      2                             *pm2

       aa(j12,j22)=aa(j12,j22) +  p(i)*p(j)*da
      2                             *pm1
      1                         +  alpha*deltat*(visc/roden)*b(1,i)
      2                            *b(1,j)*da
      2                             *pm1
      3                         +  alpha*deltat*((2.0*visc/roden)
      4                            +(1.0/roden)*bulk)
      5                            *b(2,i)*b(2,j)*da
      2                             *pm2
      5                         +  alpha*deltat*(visc/roden)*b(3,i)
```

```
     2                              *pm1

      aa(j12,j23)=aa(j12,j23) +  alpha*deltat*(visc/roden)*b(3,i)
     1                           *b(2,j)*da
     2                            *pm1
     2                        +  alpha*deltat*((1.0/roden)*bulk)
     3                           *b(2,i)*b(3,j)*da
     2                            *pm2

      aa(j13,j21)=aa(j13,j21) +  alpha*deltat*(visc/roden)*b(1,i)
     1                           *b(3,j)*da
     2                            *pm1
     2                        +  alpha*deltat*((1.0/roden)*bulk)
     3                           *b(3,i)*b(1,j)*da
     2                            *pm2

      aa(j13,j22)=aa(j13,j22) +  alpha*deltat*(visc/roden)*b(2,i)
     1                           *b(3,j)*da
     2                            *pm1
     2                        +  alpha*deltat*((1.0/roden)*bulk)
     3                           *b(3,i)*b(2,j)*da
     2                            *pm2

      aa(j13,j23)=aa(j13,j23) +  p(i)*p(j)*da
     2                            *pm1
     1                        +  alpha*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     2                            *pm1
     3                        +  alpha*deltat*(visc/roden)*b(2,i)
     4                           *b(2,j)*da
     2                            *pm1
     5                        +  alpha*deltat*((2.0*visc/roden)
     6                           +((1.0/roden)*bulk))
     7                           *b(3,i)*b(3,j)*da
     2                            *pm2


c --- Matrix on Right Hand Side -----------------------------------------------
c
c     For Transient State (Cartesian co-ordinate system)

      ak(j11,j21)=ak(j11,j21) +  p(i)*p(j)*da
     2                            *pm1
     1                        -  (1.0-alpha)*deltat*((2.0*visc/roden)
     2                           +((1.0/roden)*bulk))
     3                           *b(1,i)*b(1,j)*da
     2                            *pm2
     4                        -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     5                           *b(2,j)*da
     2                            *pm1
     6                        -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     7                           *b(3,j)*da
     2                            *pm1

      ak(j11,j22)=ak(j11,j22) -  (1.0-alpha)*deltat*(visc/roden)
     1                           *b(2,i)*b(1,j)*da
     2                            *pm1
     2                        -  (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(1,i)*b(2,j)*da
     2                            *pm2

      ak(j11,j23)=ak(j11,j23) -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     1                           *b(1,j)*da
     2                            *pm1
     2                        -  (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(1,i)*b(3,j)*da
     2                            *pm2

      ak(j12,j21)=ak(j12,j21) -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     1                           *b(2,j)*da
     2                            *pm1
     2                        -  (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(2,i)*b(1,j)*da
     2                            *pm2

      ak(j12,j22)=ak(j12,j22) +  p(i)*p(j)*da
     2                            *pm1
     1                        -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
```

```
     2                            *pm1
     3                        -  (1.0-alpha)*deltat*((2.0*visc/roden)
     4                           +((1.0/roden)*bulk))
     5                           *b(2,i)*b(2,j)*da
     2                            *pm2
     6                        -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     7                           *b(3,j)*da
     2                            *pm1

      ak(j12,j23)=ak(j12,j23) -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     1                           *b(2,j)*da
     2                            *pm1
     2                        -  (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(2,i)*b(3,j)*da
     2                            *pm2

      ak(j13,j21)=ak(j13,j21) -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     1                           *b(3,j)*da
     2                            *pm1
     2                        -  (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(3,i)*b(1,j)*da
     2                            *pm2

      ak(j13,j22)=ak(j13,j22) -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     1                           *b(3,j)*da
     2                            *pm1
     2                        -  (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(3,i)*b(2,j)*da
     2                            *pm2

      ak(j13,j23)=ak(j13,j23) +  p(i)*p(j)*da
     2                            *pm1
     1                        -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     2                            *pm1
     3                        -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     4                           *b(2,j)*da
     2                            *pm1
     5                        -  (1.0-alpha)*deltat*((2.0*visc/roden)
     6                           +((1.0/roden)*bulk))
     7                           *b(3,i)*b(3,j)*da
     2                            *pm2

c
c     Body Force Effect (for Elemental Load Vector Calculation)
c     ==========================================================

      C(j11) =C(j11) + (1.0-alpha)*deltat*p(j)*grav1*da
     2              *pm1

      C(j12) =C(j12) + (1.0-alpha)*deltat*p(j)*grav2*da
     2              *pm1

      C(j13) =C(j13) + (1.0-alpha)*deltat*p(j)*grav3*da
     2              *pm1

 6070 continue
 6020 continue

      endif

c ----'Reduced' Integration ----------------------------------------------

      lg = 0

      call gaussp(mgaus,xg,cg,ncn)

      do 6021 ig=1,mgaus
                                     g = xg(ig)
      do 6021 jg=1,mgaus
                                     h = xg(jg)
      do 6021 kg=1,mgaus
                                     f = xg(kg)

                                     lg = lg + 1


      if(iter.eq.1) then
     1                     nnp, maxel, maxst, maxnp, ndim)

      call deriv (iel,ig,jg,kg,p,del,b,ncn,da,cg,node,cord,
     1              maxel,maxnp)

                iig=ig
                jjg=jg
                kkg=kg

                write(16)   iel ,ig ,jg ,kg, p ,del ,b ,da

      else
      if(.not. EOF(16))read(16) iel,iig,jjg,kkg,p ,del ,b , da

      endif
c
c     calculation of viscosity based on the constitutive equation.
c
      spress = 0.0
      stemp  = 0.0

             do 5334 ip = 1,ncn
             jp    = iabs(node(iel,ip))
             stemp = stemp + temp(jp) * p(ip)
 5334        continue
             epsii = 1.d-10
             gamad = sinv(iel,lg)
             if(gamad.lt.epsii) gamad = epsii

      call visca
     1(rvisc,power,visc,stemp,rtem,tbco,spress,rpef,taco,gamad)

c -----calculate viscosity dependent penalty parameter---------------

      bulk = rbulk * visc

      rmat1(iel,lg) = visc

      rmat2(iel,lg) = bulk
c-------------------------------------------------------------------
c     preparation of the convective acceleration terms/balancing
c     dissipation is used
c
                 do 6051 idff= 1,3
                 x(idff)     = 0.0
                 v(idff)     = 0.0
                 hh(idff)    = 0.0
6051             continue
         do 6061 icn = 1 ,ncn
               jcn = iabs(node(iel,icn))
         do 6061 idff= 1 ,  3
         x(idff)     = x(idff) + p(icn)*cord(jcn,idff)
         v(idff)     = v(idff) + p(icn)*vel (jcn,idff)
6061     continue


      if(icord.eq.1) then
c
c     modify da for axisymmetric computations.
c
      da = da * x(1)
      endif
c
c     column index
c
      do 6071 i=1,ncn

                 j11= i
                 j12= i + ncn
                 j13= i + 2*ncn
                 j14= i + 3*ncn

      do 6071 j=1,ncn
                 j21= j
                 j22= j + ncn
                 j23= j + 2*ncn
                 j24= j + 3*ncn
```

```
c
c --- Stiffness Matrix of Left Hand Side ('reduced' integration)--------------------------
c
c      For Transient state (Cartesian co-ordinate system)


       aa(j11,j21)=aa(j11,j21) +  p(i)*p(j)*da
     2                             *(1.0-pm1)
     1                         +  alpha*deltat*((2.0*visc/roden)
     2                            +((1.0/roden)*bulk))
     3                            *b(1,i)*b(1,j)*da
     2                             *(1.0-pm2)
     3                         +  alpha*deltat*(visc/roden)*b(2,i)
     4                            *b(2,j)*da
     2                             *(1.0-pm1)
     5                         +  alpha*deltat*(visc/roden)*b(3,i)
     6                            *b(3,j)*da
     2                             *(1.0-pm1)

       aa(j11,j22)=aa(j11,j22) +  alpha*deltat*(visc/roden)
     1                            *b(2,i)*b(1,j)*da
     2                             *(1.0-pm1)
     2                         +  alpha*deltat*((1.0/roden)*bulk)
     3                            *b(1,i)*b(2,j)*da
     2                             *(1.0-pm2)

       aa(j11,j23)=aa(j11,j23) +  alpha*deltat*(visc/roden)*b(3,i)
     1                            *b(1,j)*da
     2                             *(1.0-pm1)
     2                         +  alpha*deltat*((1.0/roden)*bulk)
     3                            *b(1,i)*b(3,j)*da
     2                             *(1.0-pm2)

       aa(j12,j21)=aa(j12,j21) +  alpha*deltat*(visc/roden)*b(1,i)
     1                            *b(2,j)*da
     2                             *(1.0-pm1)
     2                         +  alpha*deltat*((1.0/roden)*bulk)
     3                            *b(2,i)*b(1,j)*da
     2                             *(1.0-pm2)

       aa(j12,j22)=aa(j12,j22) +  p(i)*p(j)*da
     2                             *(1.0-pm1)
     1                         +  alpha*deltat*(visc/roden)*b(1,i)
     2                            *b(1,j)*da
     2                             *(1.0-pm1)
     3                         +  alpha*deltat*((2.0*visc/roden)
     4                            +(1.0/roden)*bulk)
     5                            *b(2,i)*b(2,j)*da
     2                             *(1.0-pm2)
     5                         +  alpha*deltat*(visc/roden)*b(3,i)
     6                            *b(3,j)*da
     2                             *(1.0-pm1)

       aa(j12,j23)=aa(j12,j23) +  alpha*deltat*(visc/roden)*b(3,i)
     1                            *b(2,j)*da
     2                             *(1.0-pm1)
     2                         +  alpha*deltat*((1.0/roden)*bulk)
     3                            *b(2,i)*b(3,j)*da
     2                             *(1.0-pm2)

       aa(j13,j21)=aa(j13,j21) +  alpha*deltat*(visc/roden)*b(1,i)
     1                            *b(3,j)*da
     2                             *(1.0-pm1)
     2                         +  alpha*deltat*((1.0/roden)*bulk)
     3                            *b(3,i)*b(1,j)*da
     2                             *(1.0-pm2)

       aa(j13,j22)=aa(j13,j22) +  alpha*deltat*(visc/roden)*b(2,i)
     1                            *b(3,j)*da
     2                             *(1.0-pm1)
     2                         +  alpha*deltat*((1.0/roden)*bulk)
     3                            *b(3,i)*b(2,j)*da
     2                             *(1.0-pm2)

       aa(j13,j23)=aa(j13,j23) +  p(i)*p(j)*da
     2                             *(1.0-pm1)
     1                         +  alpha*deltat*(visc/roden)*b(1,i)
     2                            *b(1,j)*da
     2                             *(1.0-pm1)
     3                         +  alpha*deltat*(visc/roden)*b(2,i)
     2                            *(1.0-pm1)
     5                      +    alpha*deltat*((2.0*visc/roden)
     6                           +((1.0/roden)*bulk))
     7                           *b(3,i)*b(3,j)*da
     2                            *(1.0-pm2)

c --- Matrix on Right Hand Side ------------------------------------------------
c
c     For Transient State (Cartesian co-ordinate system)


      ak(j11,j21)=ak(j11,j21) +  p(i)*p(j)*da
     2                            *(1.0-pm1)
     1                      -    (1.0-alpha)*deltat*((2.0*visc/roden)
     2                           +((1.0/roden)*bulk))
     3                           *b(1,i)*b(1,j)*da
     2                            *(1.0-pm2)
     4                      -    (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     5                           *b(2,j)*da
     2                            *(1.0-pm1)
     6                      -    (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     7                           *b(3,j)*da
     2                            *(1.0-pm1)

      ak(j11,j22)=ak(j11,j22) -  (1.0-alpha)*deltat*(visc/roden)
     1                           *b(2,i)*b(1,j)*da
     2                            *(1.0-pm1)
     2                      -    (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(1,i)*b(2,j)*da
     2                            *(1.0-pm2)

      ak(j11,j23)=ak(j11,j23) -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     1                           *b(1,j)*da
     2                            *(1.0-pm1)
     2                      -    (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(1,i)*b(3,j)*da
     2                            *(1.0-pm2)

      ak(j12,j21)=ak(j12,j21) -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     1                           *b(2,j)*da
     2                            *(1.0-pm1)
     2                      -    (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(2,i)*b(1,j)*da
     2                            *(1.0-pm2)

      ak(j12,j22)=ak(j12,j22) +  p(i)*p(j)*da
     2                            *(1.0-pm1)
     1                      -    (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     2                            *(1.0-pm1)
     3                      -    (1.0-alpha)*deltat*((2.0*visc/roden)
     4                           +((1.0/roden)*bulk))
     5                           *b(2,i)*b(2,j)*da
     2                            *(1.0-pm2)
     6                      -    (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     7                           *b(3,j)*da
     2                            *(1.0-pm1)

      ak(j12,j23)=ak(j12,j23) -  (1.0-alpha)*deltat*(visc/roden)*b(3,i)
     1                           *b(2,j)*da
     2                            *(1.0-pm1)
     2                      -    (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(2,i)*b(3,j)*da
     2                            *(1.0-pm2)

      ak(j13,j21)=ak(j13,j21) -  (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     1                           *b(3,j)*da
     2                            *(1.0-pm1)
     2                      -    (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(3,i)*b(1,j)*da
     2                            *(1.0-pm2)

      ak(j13,j22)=ak(j13,j22) -  (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     1                           *b(3,j)*da
     2                            *(1.0-pm1)
     2                      -    (1.0-alpha)*deltat*((1.0/roden)*bulk)
     3                           *b(3,i)*b(2,j)*da
     2                            *(1.0-pm2)
```

```
      ak(j13,j23)=ak(j13,j23) +  p(i)*p(j)*da
     2                            *(1.0-pm1)
     1                      -    (1.0-alpha)*deltat*(visc/roden)*b(1,i)
     2                           *b(1,j)*da
     2                            *(1.0-pm1)
     3                      -    (1.0-alpha)*deltat*(visc/roden)*b(2,i)
     4                           *b(2,j)*da
     2                            *(1.0-pm1)
     5                      -    (1.0-alpha)*deltat*((2.0*visc/roden)
     6                           +((1.0/roden)*bulk))
     7                           *b(3,i)*b(3,j)*da
     2                            *(1.0-pm2)

c
c     Body Force Effect (for Elemental Load Vector Calculation)
c     =========================================================
c

      C(j11) =C(j11) + (1.0-alpha)*deltat*p(j)*grav1*da
     2            *(1.0-pm1)

      C(j12) =C(j12) + (1.0-alpha)*deltat*p(j)*grav2*da
     2            *(1.0-pm1)

      C(j13) =C(j13) + (1.0-alpha)*deltat*p(j)*grav3*da
     2            *(1.0-pm1)

6071  continue
6021  continue

c     For Transient State (Cartesian Co-ordinate System)
c     ==================================================
c     Term one on RHS is evaluated
c     ===========================

      do 6080 i=1,ncn

                    j11= i
                    j12= i + ncn
                    j13= i + 2*ncn
                    j14= i + 3*ncn

      do 6080 j=1,ncn
                    j21= j
                    j22= j + ncn
                    j23= j + 2*ncn
                    j24= j + 3*ncn

      nn=iabs(node(iel,j))


      akf(j11)=akf(j11) + ak(j11,j21)*vel(nn,1) +
     1                    ak(j11,j22)*vel(nn,2) +
     2                    ak(j11,j23)*vel(nn,3)

      akf(j12)=akf(j12) + ak(j12,j21)*vel(nn,1) +
     1                    ak(j12,j22)*vel(nn,2) +
     2                    ak(j12,j23)*vel(nn,3)

      akf(j13)=akf(j13) + ak(j13,j21)*vel(nn,1) +
     1                    ak(j13,j22)*vel(nn,2) +
     2                    ak(j13,j23)*vel(nn,3)

      akf(j14)=akf(j14) + ak(j14,j21)*vel(nn,1) +
     1                    ak(j14,j22)*vel(nn,2) +
     2                    ak(j14,j23)*vel(nn,3)

6080  continue

c
c    Evaluation of Elemental Load Vector
c    ===================================

      do 6085 i=1,ncn

                    j11= i
                    j12= i + ncn
                    j13= i + 2*ncn
                    j14= i + 3*ncn
c
c     For Transient State  (Cartesian Co-ordinate System)
c
```

```
      rr(j12)= rr(j12) + akf(j12) + C(j12)
      rr(j13)= rr(j13) + akf(j13) + C(j13)

6085  continue

      maxte=maxdf

      call front
     1(aa   ,rr   ,iel  ,node ,maxel,maxst,ldest,kdest,nk   ,maxfr
     2,eq   ,lhed ,khed ,kpiv ,lpiv ,jmod ,qq   ,pvkol,vel  ,rl
     3,ncod ,bc   ,nopp ,mdf  ,ndn  ,maxdf,nel  ,maxte,ntov ,lcol
     4,nell ,ntra ,icho )

      return
      end

c     ================================================================
      subroutine front
     1    (aa   ,rr   ,iel  ,nop  ,maxel,maxst,ldest,kdest,nk   ,maxfr,
     2     eq   ,lhed ,khed ,kpiv ,lpiv ,jmod ,qq   ,pvkol,vel  ,rl   ,
     3     ncod ,bc   ,nopp ,mdf  ,ndn  ,maxdf,nel  ,maxte,ntov ,lcol ,
     4     nell ,ntra ,icho )

c
c     Frontal elimination routine using diagonal pivoting
c
      implicit double precision(a-h,o-z)
      dimension aa   (maxst,maxst) ,rr   (maxst)
      dimension nop  (maxel,maxst)
      dimension ldest(maxst)       ,kdest(maxst)       ,nk   (maxst)
      dimension eq   (maxfr,maxfr) ,lhed (maxfr)       ,khed (maxfr)
      dimension kpiv (maxfr)       ,lpiv (maxfr)
      dimension jmod (maxfr)       ,qq   (maxfr)       ,pvkol(maxfr)
      dimension vel  (maxte)       ,rl   (maxdf)       ,ncod (maxdf)
      dimension bc   (maxdf)       ,nopp (maxdf)       ,mdf  (maxdf)
      dimension ndn  (maxdf)       ,press(maxdf)
c
c     nlp and ndl are the file specifiers for units 60 and 14 respectively
c     =================================================================

      nlp=60
      ndl=14
c
c     Prefront
c     ========
      nmax=maxfr
      ncrit=20
      nlarg=maxfr-10
      if(iel.eq.1) nell = 0
      if(iel.eq.1) ntra = 1
      if(ntra.eq.0) goto 6040
      nmax = maxfr
      ntra = 0
      ncrit = 20
      lfron = 0
      nlarg = nmax-10
c
c     Find last appeareance of each node
c     ==================================
      nlast = 0
      do 6010 i = 1,ntov
      do 6020 n = 1,nel
      jdn = ndn(n)
      do 6030 l = 1,jdn
      if(nop(n,l).ne.i)go to 6030
      nlast1 = n
      nlast = n
      ll = l
6030  continue
6020  continue
      if(nlast.eq.0) go to 6010
      nop(nlast,ll) = -nop(nlast,ll)
      nlast = 0
6010  continue
      ntrix = jdn
c
c     Assembly
c     ========
6040  continue
      if(iel.gt.1) go to 6060
```

```
      lcol = 0
      do 6050 i = 1,nmax
      do 6050 j = 1,nmax
      eq(j,i) = 0.
6050  continue
6060  nell = nell+1
      n = nell
      jdn = ndn(nell)
      kc = 0
      do 6070 j = 1,jdn
      nn = nop(n,j)
      m = iabs(nn)
      k = nopp(m)
      idf = mdf(m)
      rl(m) = rr(j)+rl(m)
      do 6070 l = 1,idf
      kc = kc+1
      ii = k+l-1
      if(nn.lt.0)ii = -ii
      nk(kc) = ii
6070  continue
c
c     Set up heading vectors
c     ======================
c
      do 6080 lk = 1,kc
      node = nk(lk)
      if(lcol.eq.0)goto 6100
      do 6090 l = 1,lcol
      ll = l
      if(iabs(node).eq.iabs(lhed(l)))go to 6110
6090  continue
6100  lcol = lcol+1
      ldest(lk) = lcol
      lhed(lcol) = node
      go to 6080
6110  ldest(lk) = ll
      lhed(ll) = node
6080  continue
      if(lcol.le.nmax)go to 6130
      nerror = 2
      write(nlp,3010)nerror
      stop
6130  continue
      do 6140 l = 1,kc
      ll = ldest(l)
      do 6140 k = 1,kc
      kk = ldest(k)
      eq(kk,ll) = eq(kk,ll)+aa(k,l)
6140  continue
      if(lcol.lt.ncrit.and.nell.lt.nel) return
c
c     Find out which matrix elements are fully assembeled
c     ===================================================
6150  lc = 0
      ir = 0
      do 6160 l = 1,lcol
      kt = lhed(l)
      if(kt.ge.0)go to 6160
      lc = lc+1
      lpiv(lc) = l
      kro = iabs(kt)
      if(ncod(kro).ne.1)go to 6160
      ir = ir+1
      jmod(ir) = l
      ncod(kro) = 2
      rl(kro) = bc(kro)
6160  continue
c
c     Modify equations with applied boundary conditions
c     =================================================
      if(ir.eq.0)go to 6190
      do 6170 irr = 1,ir
      k = jmod(irr)
      kh = iabs(lhed(k))
      do 6180 l = 1,lcol
      eq(k,l) = 0.
      lh = iabs(lhed(l))
      if(lh.eq.kh)eq(k,l) = 1.
6180  continue
6170  continue
```

```
      if(lc.gt.0)go to 6200
      ncrit = ncrit+10
      if(ncrit.le.nlarg) return
      nerror = 3
      write(nlp,3030)nerror
      stop
 6200 continue
c
c     Search for absolute pivot
c     =========================
      pivot = 0.
      do 6210 l = 1,lc
      lpivc = lpiv(l)
      kpivr = lpivc
      piva = eq(kpivr,lpivc)
      if(abs(piva).lt.abs(pivot))go to 6220
      pivot = piva
      lpivco = lpivc
      kpivro = kpivr
 6220 continue
 6210 continue
      if(pivot.eq.0.0) return
c
c     Normalise pivotal row
c     =====================
      lco = iabs(lhed(lpivco))
      kro = lco

      if(abs(pivot).lt.0.1d-28) write(nlp,3050)
      do 6240 l = 1,lcol
      qq(l) = eq(kpivro,l)/pivot
 6240 continue
      rhs = r1(kro)/pivot
      r1(kro) = rhs
      pvkol(kpivro) = pivot
c
c     Eliminate then delete pivotal row and column
c     ============================================
      if(kpivro.eq.1)go to 6300
      kpivr = kpivro-1
      do 6250 k = 1,kpivr
      krw = iabs(lhed(k))
      fac = eq(k,lpivco)
      pvkol(k) = fac
      if(lpivco.eq.1.or.fac.eq.0.)go to 6270
      lpivc = lpivco-1
      do 6260 l = 1,lpivc
      eq(k,l) = eq(k,l)-fac*qq(l)
 6260 continue
 6270 if(lpivco.eq.lcol)go to 6290
      lpivc = lpivco+1
      do 6280 l = lpivc,lcol
      eq(k,l-1) = eq(k,l)-fac*qq(l)
 6280 continue
 6290 r1(krw) = r1(krw)-fac*rhs
 6250 continue
 6300 if(kpivro.eq.lcol)go to 6360
      kpivr = kpivro+1
      do 6310 k = kpivr,lcol
      krw = iabs(lhed(k))
      fac = eq(k,lpivco)
      pvkol(k) = fac
      if(lpivco.eq.1)go to 6330
      lpivc = lpivco-1
      do 6320 l = 1,lpivc
      eq(k-1,l) = eq(k,l)-fac*qq(l)
 6320 continue
 6330 if(lpivco.eq.lcol)go to 6350
      lpivc = lpivco+1
      do 6340 l = lpivc,lcol
      eq(k-1,l-1) = eq(k,l)-fac*qq(l)
 6340 continue
 6350 r1(krw) = r1(krw)-fac*rhs
 6310 continue
 6360 continue
c
c     Write pivotal equation on disc
c     ==============================
      write(nd1) kro,lcol,lpivco,(lhed(l),qq(l),l = 1,lcol)
      do 6370 l = 1,lcol
```

```
      eq(l,lcol) = 0.
      eq(lcol,l) = 0.
 6370 continue
c
c     Rearrange heading vectors
c     =========================
      lcol = lcol-1
      if(lpivco.eq.lcol+1)go to 6390
      do 6380 l = lpivco,lcol
      lhed(l) = lhed(l+1)
 6380 continue
 6390 continue
c
c     Determine whether to assemble,eliminate,or backsubstitute
c     =========================================================
      if(lcol.gt.ncrit)go to 6150
      if(nell.lt.nel) return
      if(lcol.gt.1)go to 6150
      lco = iabs(lhed(1))
      kpivro = 1
      pivot = eq(1,1)
      kro = lco
      lpivco = 1
      qq(1) = 1.

      if(abs(pivot).lt.1d-28)go to 6410

      r1(kro) = r1(kro)/pivot
      write(nd1) kro,lcol,lpivco,lhed(1),qq(1)
c
c     start back-substitution
c     =======================
      call bacsub
     1    (ntov ,ncod ,bc   ,r1   ,vel  , maxfr,qq   ,lhed ,nd1,
     2     icho)


c     main exit with solution
c     =======================
 6410 continue

 3010 format(/'  nerror=',i5//
     1 '  the difference nmax-ncrit is not sufficiently large'
     1/'  to permit the assembly of the next element---'
     1/'  either increase nmax or lower ncrit'
     1/)
 3030 format(/'  nerror=',i5//
     1 '  there are no more rows fully summed,this may be due to---'
     1/'  (1)incorrect coding of nop or nk arrays'
     1/'  (2)incorrect value of ncrit. increase ncrit to permit'
     1/'     whole front to be assembled'
     1/)

 3050 format('  warning-matrix singular or ill conditioned')

      return
      end

c     ==================================================================
      subroutine gaussp(ngaus,xg,cg,ncn)

      implicit double precision(a-h,o-z)
c
c     x(g)   specifies the coordinates of the Gauss points
c     c(g)   specifies the Gauss weights
c
      dimension xg(3),cg(3)

      if(ngaus.eq.1) then
      xg(1)=0.0
      cg(1)=2.0

      elseif(ngaus.eq.2)  then
      xg(1) = 0.57735026919d00
      xg(2) = -xg(1)
      cg(1) = 1.00
      cg(2) = 1.00

      else
      xg(1) = 0.77459666924d00
      xg(2) = 0.0
      cg(1) = 0.55555555556d00
      cg(2) = 0.88888888889d00
      cg(3) = cg(1)

      endif
      return
      end

c     ==================================================================
      subroutine getnod (nnp  ,cord ,idv1 ,idv2 ,maxnp,ndim,icord)

      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     nnp    total number of nodal points in the mesh
c     cord   array for nodal coordinates
c     idv1   input device id.
c     idv2   output device id.
c     ndim   see below
c
      dimension cord(maxnp, ndim)

      if (.NOT. EOF(51)) read (idv1,1010) (jnp ,(cord(jnp,idf),idf=1,3)
     1    ,jnp=1,nnp)
      if(icord.eq.0) write(idv2,3010)
      if(icord.eq.1) write(idv2,3020)
      write(idv2,3030) (jnp ,(cord(jnp,idf),idf=1,3) ,jnp=1,nnp)

      return

1010  format(i7,e20.12,e20.12,e20.12)
3010  format(' ',///' ',20('*'),' nodal coordinates ',20('*'),//
     1' ',(7x,'id.',13x,'x-coord',13x,'y-coord',13x,'z-coord',13x)/)
3020  format(' ',///' ',20('*'),' nodal coordinates ',20('*'),//
     1' ',2(7x,'id/',7x,'r-coord',7x,'z-coord',20x)/)
3030  format(' ',i10,10x,f10.6,10x,f10.6,10x,f10.6)

      end

c     ===================================================================
      subroutine getelm (nel  ,ncn  ,node ,idv1 ,idv2 ,maxel)

      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     ncn    number of nodes per element
c     node   array for element connectivity data
c     idv1   input device id.
c     idv2   output device id.
c     maxel  see below

      dimension node (maxel, ncn)

              do 6010 iel = 1 ,nel
6010  if (.not. eof(51))read (idv1,1010) iel ,(node(iel,icn),icn=1,ncn)
      print*, "nodal connectivity array read"
      write(idv2,3010)
               do 6020 jel = 1 ,nel
6020  write(idv2,3020) jel ,(node(jel,icn),icn=1,ncn)

      return

1010  format(21i7)
3010  format(' ',///,' ',20('*'),' element connectivity ',20('*'),//
     1' ',4x,'id.',7x,'nodal-point entries',/)
3020  format(21i7)

      end

c     ===================================================================c
      subroutine getbcd (nbc  ,ibc  ,jbc  ,vbc
     1                    ,idv1 ,idv2 ,maxbc)

      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     nbc    number of nodal constraint data
```

```
c     ibc    array for constrained nodal points
c     jbc    array for constrained degree of freedom
c     vbc    array for boundary values
c     udv1   input device id.
c     idv2   output device id.
c     maxbc  see below
c
      dimension ibc  (maxbc) ,jbc  (maxbc),vbc  (maxbc)

      if (.not. eof(51))read (idv1,1010) (ibc(ind) ,jbc(ind) ,vbc(ind)
     1    ,ind=1,nbc)
      print*, "boundary conditions array read"
      write(idv2,3010)
      write(idv2,3020) (ibc(ind) ,jbc(ind) ,vbc(ind) ,ind=1,nbc)

      return

1010  format(2i5,f10.4)
3010  format(' ',//  /,' ',20('*'),' nodal constraint ',20('*'),//
     1' ',(8x,'id.',7x,'dof',10x,'value',10x)/)
3020  format(5x,i5,5x,i5,f17.4)

      end

c     ======================================================================

      subroutine lumpm
     1    (clump,nnp  ,maxnp,nel  ,ngaus,p   ,del  ,b   ,maxst,
     2     node ,maxel,ncn  )

      implicit double precision(a-h,o-z)
      dimension b    (    3,    20) ,del  (    3,    20) ,p    (  20)
      dimension clump(maxnp      )
      dimension node (maxel,maxst)
      dimension pp   (ncn  ,ncn  )

                                       do 5000 inp = 1 ,nnp
                                       clump  (inp)= 0.0
 5000                                  continue

              rewind 15

      do 5010 iel = 1 ,nel

             do 5020             ig  = 1 ,ngaus
             do 5020             jg  = 1 ,ngaus

      if(.not. EOF(15)) read (15) jel ,iig ,jjg ,kkg ,p ,del ,b ,da

                     do 5030    icn = 1 ,ncn
                                 ww = 0.0
                     do 5040    jcn = 1 ,ncn
                                 ww = ww + p(icn)*p(jcn)*da
 5040                continue
                     inp              = iabs(node(iel,icn))
                     clump(inp)       = clump(inp) + ww

 5030                continue

 5020        continue
 5010        continue

      return
      end

c     =====================================================================
      subroutine minimax
     1( cmax  , pmax  , vel   , press ,  maxnp,  nnp  ,     nc,
     2  np    , nm    , ncm   , nvxm  ,  nvym  ,  nvzm ,
     3  nvxl  , nvyl  , nvzl  , pmin  ,  cmin  ,
     4  vxmax , vxmin , vymax, vymin ,  vzmax ,  vzmin,  ndim , maxdf,
     5  stres )
      implicit real*8 (a-h,o-z)
      dimension vel   (maxdf)
      dimension press ( maxnp )   , stres(maxnp, 11)

      vxmax   =   vel(1)
      vxmin   =   vel(1)
      vymax   =   vel(nnp+1)
      vymin   =   vel(nnp+1)
      vzmax   =   vel(2*nnp+1)
      pmax    =   stres (1,7)
      pmin    =   stres (1,7)

      nc      =   1
      ncm     =   1
      np      =   1
      nm      =   1
      nvxm        =   1
      nvym        =   1
      nvzm        =   1
      nvxl        =   1
      nvyl        =   1
      nvzl        =   1

      do 6020 i=2,nnp

      pm      =   stres (i,7)
      pi      =   stres (i,7)
      vxmx    =   vel(i)
      vxmn    =   vel(i)
      vymx    =   vel(nnp+i)
      vymn    =   vel(nnp+i)
      vzmx    =   vel(2*nnp+i)
      vzmn    =   vel(2*nnp+i)

         if (   pm.gt.pmax  ) then
            pmax=pm
            np  =i
         endif
         if (   pi.lt.pmin  ) then
            pmin = pi
            nm   = i
         endif
         if ( vxmx.gt.vxmax ) then
            vxmax= vxmx
            nvxm  = i
         endif
         if ( vymx.gt.vymax ) then
            vymax= vymx
            nvym  = i
         endif
         if ( vzmx.gt.vzmax ) then
            vzmax= vzmx
            nvzm  = i
         endif

         if ( vxmn.lt.vxmin ) then
            vxmin= vxmn
            nvxl  = i
         endif
         if ( vymn.lt.vymin ) then
            vymin= vymn
            nvyl  = i
         endif
         if ( vzmn.lt.vzmin ) then
            vzmin= vzmn
            nvzl  = i
         endif
 6020 continue

      return
      end

c     ==================================================================
      subroutine putbcv
     1 (nnp  ,nbc ,ibc ,jbc ,vbc ,ncod ,bc ,maxbc,maxdf,maxel,maxst,
     2  node)

      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     ncod    array for constraint switch defined for every d.o.f.
c     bc      array for storing contraint value
c     maxbc   see below
c     maxdf   see below
c
      dimension ibc  (maxbc) ,jbc  (maxbc) ,vbc  (maxbc)
      dimension ncod (maxdf) ,bc   (maxdf) ,node (maxel,maxst)
```

```
      do 6010 ind = 1 ,nbc
      if(jbc(ind)>3) goto 6010
      jnd           = ibc(ind)+(jbc(ind)-1)*nnp
      bc    (jnd) = vbc(ind)
      ncod  (jnd) =  1
6010  continue

      return
      end

c    ======================================================

      subroutine setprm
     1      (nnp  ,nel  ,ncn  ,node ,ndf  ,maxel,maxst,ndn  ,ntrix,
     2       maxdf,ntov ,mdf  ,nopp )

      implicit double precision(a-h,o-z)

c     arguments
c     =========
c     arguments are already defined
c
      dimension node (maxel,maxst), ndn  (maxdf)
      dimension mdf  (maxdf      ), nopp (maxdf)
c
c     function
c     ========
c     Sets the location data for nodal degrees of freedom
c
      do 6010 iel = 1 ,nel
         ndn(iel) = ntrix
      do 6010 icn = 1 ,ncn
              kcn = node(iel,icn)

              jbcn= icn+(ndf-2)*ncn
              lbcn= kcn+(ndf-2)*nnp

              jccn= icn+(ndf-1)*ncn
              lccn= kcn+(ndf-1)*nnp

      node(iel,jbcn) = lbcn
      node(iel,jccn) = lccn

6010  continue
      do 6020 idf = 1,ntov
         mdf(idf) = 1
        nopp(idf) = idf
6020  continue

      return
      end
c
c     ==================================================================
      subroutine getmat (nel ,nmat,pmat, idv1, idv2,maxel,rtem, rpef)
c
      implicit double precision(a-h,o-z)
c
c     arguments
c     =========
c     nmat    number of materials
c     pmat    array for material constants for each element
c     idv1    input device id.
c     idv2    output devide id.
c     roden    density
c     rvisc    mu nought; consistency coefficient
c     pref     reference pressure
c     power    power law index
c     tref     reference temperature
c     tbco     coefficient b in the power law model
c     taco     coefficient a in the power law model
c     dispc    dispersion coefficient
c     gamad    shear rate
c
      dimension pmat (maxel,  10)

      write(idv2,3010)

      do 6010 imat = 1 ,nmat
      if (.NOT. EOF(51)) read(idv1,1010) rvisc, power, tref, tbco, taco,
     1                                   dispc, pref, roden, gamad,rbulk
      print*, 'material properties read'
```

```
             ito    = nel

             if(rtem .eq.0.) rtem  = 0.001
             if(rpef .eq.0.) rpef  = 0.001

      do 6020 iel  = ifrom ,ito
         pmat(iel,1) = rvisc
         pmat(iel,2) = pref
         pmat(iel,3) = power
         pmat(iel,4) = tref
         pmat(iel,5) = tbco
         pmat(iel,6) = taco
         pmat(iel,7) = dispc
         pmat(iel,8) = roden
         pmat(iel,9) = gamad
         pmat(iel,10)= rbulk
         rtem        = tref
         rpef        = pref

6020  continue

      write(idv2,3020) imat ,ifrom ,ito ,rvisc ,power
      write(idv2,3030)
      write(idv2,3040) tref ,tbco, pref, taco
      write(idv2,3050)
      write(idv2,3060) dispc , roden , gamad
      write(idv2,3070)
      write(idv2,3080) rbulk

6010  continue

      return

1010  format(10d10.5)

3010  format(' ',//' ',35('*'),' material properties ',35('*'),//
     1     ' ',2x,'id.',5x,'eid.(from-to)',3x,'consistency co-efficient'
     2,5x,'power law index',/)
3020  format(' ',i3,i12,i4,5x,g15.5,15x,g15.5)
3030  format(/x,' reference temperature   coefficient b
     1   reference pressure  coefficient a '/)
3040  format(f16.3,f22.4,6x,g10.3,9x,g10.3)
3050  format(/x,
     1'Dispersion Coefficient     Density           Shear rate'/)
3060  format(g13.3,15x,g7.1,6x,g16.5)
3070  format(/x,
     1'Bulk Modulus'/)
3080  format(g16.5)

      end

c    ====================================================================

      subroutine output
     1     (nnp  ,vel   ,press, maxdf,maxnp,icord, stres)

      implicit double precision(a-h,o-z)
c
c     arguments are already defined
c     =============================
      dimension vel(maxdf), press(maxnp)
      dimension stres(maxnp,   11)

      write(60,3010)

      if(icord.eq.0) write(60,3020)
      if(icord.eq.1) write(60,3030)

      do 6010 inp = 1,nnp
                        jnp = inp + nnp
                        knp = inp + (2*nnp)
      press(inp)=press(inp)
      write(60,3040)inp,vel(inp),vel(jnp),vel(knp),stres(inp, 7),
     1stres(inp,1)
6010  continue

      call minimax
     1( cmax   , pmax   , vel   , press ,   maxnp,   nnp  ,     nc,
     2  np     , nm     , ncm   , nvxm  ,   nvym   ,   nvzm ,
     3  nvxl   , nvyl   , nvzl  , pmin  ,   cmin   ,
```

```
     4  vxmax , vxmin , vymax, vymin ,  vzmax ,  vzmin,  ndim , maxdf,
     5  stres )

      write(60,3045)
      write(60,3050)nvxm,vel(nvxm),nvxl,vel(nvxl)

      write(60,3055)
      write(60,3060)nvym,vel(nnp+nvym),nvyl,vel(nnp+nvyl)

      write(60,3065)
      write(60,3070)nvzm,vel(2*nnp+nvzm),nvzl,vel(2*nnp+nvzl)

      write(60,3075)
      write(60,3080)np,stres(np,7),nm,stres(nm,7)


3010  format(/' nodal velocities and  pressures  '/)
3020  format('  id.      ux          uy            uz      press  stress'/)
3030  format('  id.      ur          uz            uz                press'/)
3040  format(i5,3e13.4,e22.8,g15.5)

3045  format('node no.          max ux    node no.          min ux')
3050  format(i5,e22.8,i5,e22.8,/)

3055  format('node no.          max uy    node no.          min uy')
3060  format(i5,e22.8,i5,e22.8,/)

3065  format('node no.          max uz    node no.          min uz')
3070  format(i5,e22.8,i5,e22.8,/)

3075  format('node no.          max p     node no.          min p')
3080  format(i5,e22.8,i5,e22.8,/)

      return
      end

c    ========================================================
      subroutine secinv
     1   (nel  ,nnp   ,ncn   ,ngaus,node ,sinv ,cord ,p   ,b ,
     2    del  ,da    ,vel   ,maxnp,maxel,maxst,ndim ,icord,
     3    maxdf,num)

      implicit double precision(a-h,o-z)
c
c     function
c     --------
c     calculates the second invariant of rate of deformation
c     tensor at integration points.
c
      dimension vel  (maxnp, ndim) ,cord (maxnp, ndim)
      dimension node (maxel,maxst) ,sinv (maxel,    35)
      dimension p    (     20      ) ,del  (    3,    20)
      dimension b    (     3,    20)
                                            rewind 15
      mgaus    = ngaus - 1

      do 5000 iel= 1 , nel
              lg = 0

      do 5010 ig = 1 ,ngaus
      do 5010 jg = 1 ,ngaus
      do 5010 kg = 1 ,ngaus

              lg = lg+1

           read (15) iiel,iig,jjg,kkg,p,del,b,da

                  u11 = 0.0
                  u12 = 0.0
                  u13 = 0.0
                  u21 = 0.0
                  u22 = 0.0
                  u23 = 0.0
                  u31 = 0.0
                  u32 = 0.0
                  u33 = 0.0

         do 5020 icn = 1 ,ncn
                             jcn = iabs(node(iel,icn))

c *** components of the rate of deformation tensor
```

```
                  u11 = u11 + b(1,icn)*vel(jcn,1)
                  u12 = u12 + b(2,icn)*vel(jcn,1)
                  u13 = u13 + b(3,icn)*vel(jcn,1)
                  u21 = u21 + b(1,icn)*vel(jcn,2)
                  u22 = u22 + b(2,icn)*vel(jcn,2)
                  u23 = u23 + b(3,icn)*vel(jcn,2)
                  u31 = u31 + b(1,icn)*vel(jcn,3)
                  u32 = u32 + b(2,icn)*vel(jcn,3)
                  u33 = u33 + b(3,icn)*vel(jcn,3)

 5020    continue
c
c *** second invariant of the rate of deformation tensor
c
      sinv(iel,lg)=0.125*((u11+u11)*(u11+u11)+
     1                    (u12+u21)*(u12+u21)+
     2                    (u13+u31)*(u13+u31)+
     3                    (u21+u12)*(u21+u12)+
     4                    (u22+u22)*(u22+u22)+
     5                    (u23+u32)*(u23+u32)+
     6                    (u31+u13)*(u31+u13)+
     7                    (u32+u23)*(u32+u23)+
     8                    (u33+u33)*(u33+u33))

 5010 continue
 5000 continue

      return
      end
c
c     ===========================================================
      subroutine shape ( xi , eta , zeta, p ,del , ncn, node, cord
     1                  nnp, maxel, maxst, maxnp, ndim)
      implicit double precision (a-h,o-z)

      DIMENSION p(20) ,del(3,20)
      Dimension cord(maxnp, ndim)
      dimension node (maxel,maxst)

      if (ncn.eq.8) then
         del(1,1)=-0.125*(1-eta)*(1-zeta)
         del(1,2)=-0.125*(1+eta)*(1-zeta)
         del(1,3)= 0.125*(1+eta)*(1-zeta)
         del(1,4)= 0.125*(1-eta)*(1-zeta)
         del(1,5)=-0.125*(1-eta)*(1+zeta)
         del(1,6)=-0.125*(1+eta)*(1+zeta)
         del(1,7)= 0.125*(1+eta)*(1+zeta)
         del(1,8)= 0.125*(1-eta)*(1+zeta)
c...................................................
         del(2,1)=-0.125*(1-xi)*(1-zeta)
         del(2,2)= 0.125*(1-xi)*(1-zeta)
         del(2,3)= 0.125*(1+xi)*(1-zeta)
         del(2,4)=-0.125*(1+xi)*(1-zeta)
         del(2,5)=-0.125*(1-xi)*(1+zeta)
         del(2,6)= 0.125*(1-xi)*(1+zeta)
         del(2,7)= 0.125*(1+xi)*(1+zeta)
         del(2,8)=-0.125*(1+xi)*(1+zeta)
c.......................................................
         del(3,1)=-0.125*(1-xi)*(1-eta)
         del(3,2)=-0.125*(1-xi)*(1+eta)
         del(3,3)=-0.125*(1+xi)*(1+eta)
         del(3,4)=-0.125*(1+xi)*(1-eta)
         del(3,5)= 0.125*(1-xi)*(1-eta)
         del(3,6)= 0.125*(1-xi)*(1+eta)
         del(3,7)= 0.125*(1+xi)*(1+eta)
         del(3,8)= 0.125*(1+xi)*(1-eta)
c.......................................................

         p(1)=0.125*(1-xi)*(1-eta)*(1-zeta)
         p(2)=0.125*(1-xi)*(1+eta)*(1-zeta)
         p(3)=0.125*(1+xi)*(1+eta)*(1-zeta)
         p(4)=0.125*(1+xi)*(1-eta)*(1-zeta)
         p(5)=0.125*(1-xi)*(1-eta)*(1+zeta)
         p(6)=0.125*(1-xi)*(1+eta)*(1+zeta)
         p(7)=0.125*(1+xi)*(1+eta)*(1+zeta)
         p(8)=0.125*(1+xi)*(1-eta)*(1+zeta)
c.......................................................

      endif
```

```
.............................................................

      endif
      return
      end

c     ================================================================
      subroutine stress
     1     (nel,nnp,ncn ,node ,p , b , da ,vel  ,maxnp, maxel,  maxst ,
     2      maxdf, stres, press, rvisc ,clump ,ngaus, rmat1,rmat2  )

       implicit double precision(a-h,o-z)

c     function
c     --------
c     calculates stress components at integration points,
c     Direct Approach (coupled scheme)
c
      dimension node (maxel,maxst)  ,  b    (  3,  20) ,cord (maxnp, 3)
      dimension stres(maxnp,   11)  ,  press(maxnp   ) ,p (  20    )
      dimension vel  (maxnp,    3)  ,  del  (  3,  20)
      dimension rmat1(maxel,   35)  ,  rmat2(maxel, 35)
      dimension clump(maxnp      )
                                                            rewind 16

            do 4990 inp   =1,maxnp
            do 4990 icp   =1, 11
            stres(inp,icp)= 0.0
 4990       continue

      mgaus    = ngaus - 1
      ired     = 1

      do 5000 iel = 1 ,nel

      ng = 0

      do 6010 ig=1,mgaus

      do 6010 jg=1,mgaus

      do 6010 kg=1,mgaus

      ng = 1 + ng

       if(.not. EOF(16))read(16) iiel,iig,jjg,kkg,p ,del ,b , da

      ifg  = ng

      rvisc=rmat1(iel,ifg)
      rbulk=rmat2(iel,ifg)

                  u11 = 0.0
                  u12 = 0.0
                  u13 = 0.0
                  u21 = 0.0
                  u22 = 0.0
                  u23 = 0.0
                  u31 = 0.0
                  u32 = 0.0
                  u33 = 0.0

        do 6020 icn = 1 ,ncn
                                   jcn = iabs(node(iel,icn))
                  u11 = u11 + b(1,icn)*vel(jcn,1)
                  u12 = u12 + b(2,icn)*vel(jcn,1)
                  u13 = u13 + b(3,icn)*vel(jcn,1)
                  u21 = u21 + b(1,icn)*vel(jcn,2)
                  u22 = u22 + b(2,icn)*vel(jcn,2)
                  u23 = u23 + b(3,icn)*vel(jcn,2)
                  u31 = u31 + b(1,icn)*vel(jcn,3)
                  u32 = u32 + b(2,icn)*vel(jcn,3)
                  u33 = u33 + b(3,icn)*vel(jcn,3)

 6020   continue

c    cartesian components of the stress tensor
c    =========================================
c    Shear Stress (Tau)
c    =========================================
      sd11 = 2.0   *rvisc *  u11
      sd22 = 2.0   *rvisc *  u22
      sd33 = 2.0   *rvisc *  u33
      sd12 = rvisc * (u12 +  u21)
      sd13 = rvisc * (u13 +  u31)
      sd23 = rvisc * (u23 +  u32)

c    =========================================
c    Normal Stress (Pi)
c    =========================================

      s11  =-pres + sd11
      s22  =-pres + sd22
      s33  =-pres + sd33
      s12  = sd12
      s13  = sd13
      s23  = sd23

c    ================================================================
c *** calculate stress at nodal points (Variational Recovery)
c    ================================================================

         do 6500 icn = 1 ,ncn
                                    jcn = iabs(node(iel,icn))

                    stres(jcn,1)= stres(jcn,1)
     1                          + p(icn) *s11  *da /clump(jcn)
                    stres(jcn,2)= stres(jcn,2)
     1                          + p(icn) *s22  *da /clump(jcn)
                    stres(jcn,3)= stres(jcn,3)
     1                          + p(icn) *s33  *da /clump(jcn)
                    stres(jcn,4)= stres(jcn,4)
     1                          + p(icn) *s12  *da /clump(jcn)
                    stres(jcn,5)= stres(jcn,5)
     1                          + p(icn) *s13  *da /clump(jcn)
                    stres(jcn,6)= stres(jcn,6)
     1                          + p(icn) *s23  *da /clump(jcn)
                    stres(jcn,7)= stres(jcn,7)
     1                          + p(icn) *pres *da /clump(jcn)
 6500               continue

 6010 continue

 5000 continue

      write(17,2100)
 2100 format(//,' Nodal Stress ',//
     1'  node',7x,'s11',12x,'s22',12x,'s33',12x,
     2's12',12x,'s13',12x,'s23'/)

      write(17,2110) (inp,(stres(inp,icp),icp=1,7),inp=1,nnp)
 2110 format(' ',i5,7g15.5)

      return

      end
c
c     ==================================================================
      subroutine visca
     1          (rvisc,power,visc,stemp,rtem,tbco,spress,rpef,taco
     2          ,gamad )

      implicit double precision(a-h,o-z)

        visc    = rvisc*(4.0*gamad**((power-1.0)*0.5))
     1            *exp(-tbco*(stemp-rtem))

      return
      end

c     ==================================================================
c     e    n    d    o    f    p    r    o    g    r    a    m
c     ==================================================================
```

APPENDIX 5

Sample Input File

3DFEANOF

– U version

3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow – UVWP version

Sample Input File

For a domain discretized into 8-noded prism element

Kee Chien Ting

Advanced Separation Techniques Group
Department of Chemical Engineering
Loughborough University

```
3D Stokes, Sample Input File
   8    3
 6560 4608 3363    1
   1    0
      0.0         0.0         0.0
   0.00001    0.00001    0.00001    0.00001
   80.0D00    1.30D00 298.00D00    0.00D00    2.00D-1    2.00D-1    1.013D5
1.000D04    2.00D-1  2.930D03    0.19D00
      1       0.51000002E-02        0.54000001E-02        0.54500001E-02
      2       0.51374999E-02        0.54000001E-02        0.54500001E-02
      3       0.51750001E-02        0.54000001E-02        0.54500001E-02
      4       0.52125002E-02        0.54000001E-02        0.54500001E-02
      5       0.52500004E-02        0.54000001E-02        0.54500001E-02
      6       0.52875001E-02        0.54000001E-02        0.54500001E-02
      7       0.53250003E-02        0.54000001E-02        0.54500001E-02
      8       0.53625004E-02        0.54000001E-02        0.54500001E-02
      9       0.54000001E-02        0.54000001E-02        0.54500001E-02
     10       0.51000002E-02        0.54000001E-02        0.54142857E-02
      .
      .
      .
   6550       0.52875001E-02        0.50499998E-02        0.49500000E-02
   6551       0.52875001E-02        0.50250003E-02        0.49500000E-02
   6552       0.52875001E-02        0.49999999E-02        0.49500000E-02
   6553       0.53250003E-02        0.50750002E-02        0.49500000E-02
   6554       0.53250003E-02        0.50499998E-02        0.49500000E-02
   6555       0.53250003E-02        0.50250003E-02        0.49500000E-02
   6556       0.53250003E-02        0.49999999E-02        0.49500000E-02
   6557       0.53625004E-02        0.50750002E-02        0.49500000E-02
   6558       0.53625004E-02        0.50499998E-02        0.49500000E-02
   6559       0.53625004E-02        0.50250003E-02        0.49500000E-02
   6560       0.53625004E-02        0.49999999E-02        0.49500000E-02
      1       2       1     136     137      11      10     145     146
      2       3       2     137     138      12      11     146     147
      3       4       3     138     139      13      12     147     148
      4       5       4     139     140      14      13     148     149
      5       6       5     140     141      15      14     149     150
      6       7       6     141     142      16      15     150     151
      7       8       7     142     143      17      16     151     152
      8       9       8     143     144      18      17     152     153
      9      11      10     145     146      20      19     154     155
     10      12      11     146     147      21      20     155     156
      .
      .
      .
   4600    6517    6516    6551    6552    6522    6521    6555    6556
   4601    6519    6518    1213    6553    6524    6523    1214    6557
   4602    6520    6519    6553    6554    6525    6524    6557    6558
   4603    6521    6520    6554    6555    6526    6525    6558    6559
   4604    6522    6521    6555    6556    6527    6526    6559    6560
   4605    6524    6523    1214    6557    6529    6528    1215    6265
   4606    6525    6524    6557    6558    6530    6529    6265    6260
   4607    6526    6525    6558    6559    6531    6530    6260    6255
   4608    6527    6526    6559    6560    6532    6531    6255    6250
   1    1    0.00000
   2    1    0.00000
   3    1    0.00000
   4    1    0.00000
   5    1    0.00000
   6    1    0.00000
   7    1    0.00000
```

```
    8     1    0.00000
    9     1    0.00000
    .
    .
    .
    1     2    0.00000
    2     2    0.00000
    3     2    0.00000
    4     2    0.00000
    5     2    0.00000
    6     2    0.00000
    7     2    0.00000
    8     2    0.00000
    9     2    0.00000
    .
    .
    .
    1     3   -0.00010
    2     3   -0.00010
    3     3   -0.00010
    4     3   -0.00010
    5     3   -0.00010
    6     3   -0.00010
    7     3   -0.00010
    8     3   -0.00010
    9     3   -0.00010

---------END OF FILE-------------------------------------------
```

The purpose of this file is to display the format of the input file and is not intended to be comprehensive. The numbers are random and due to practical reasons, a portion of the data has been truncated and has been indicated with '....'.

3DFEANOF

– P version

3-**D**imensional **F**inite **E**lement **A**nalysis of
No**n**-Newtonian fluid **F**low – Continuous **P**enalty version

Sample Input File

For a domain discretized into 8-noded prism element

Kee Chien Ting

Advanced Separation Techniques Group
Department of Chemical Engineering
Loughborough University

```
3D Stokes, Sample Input File, Penalty
    8    3
  7778 5560 3841    1
    1    0
       0.0        0.0        0.0
   0.00001    0.00001    0.00001    0.00001
   80.0D00    1.30D00 298.00D00    0.00D00    2.00D-1    2.00D-1    1.013D5
0.970D03   2.00D-1   10.0D20
       1       0.51000002E-02       0.55000000E-02       0.54500001E-02
       2       0.51399996E-02       0.55000000E-02       0.54500001E-02
       3       0.51799999E-02       0.55000000E-02       0.54500001E-02
       4       0.52200002E-02       0.55000000E-02       0.54500001E-02
       5       0.52600000E-02       0.55000000E-02       0.54500001E-02
       6       0.53000003E-02       0.55000000E-02       0.54500001E-02
       7       0.53399997E-02       0.55000000E-02       0.54500001E-02
       8       0.53800000E-02       0.55000000E-02       0.54500001E-02
       9       0.54200003E-02       0.55000000E-02       0.54500001E-02
      10       0.54600001E-02       0.55000000E-02       0.54500001E-02
       .
       .
       .
    7770       0.53800000E-02       0.49999999E-02       0.49500000E-02
    7771       0.54200003E-02       0.50750002E-02       0.49500000E-02
    7772       0.54200003E-02       0.50499998E-02       0.49500000E-02
    7773       0.54200003E-02       0.50250003E-02       0.49500000E-02
    7774       0.54200003E-02       0.49999999E-02       0.49500000E-02
    7775       0.54600001E-02       0.50750002E-02       0.49500000E-02
    7776       0.54600001E-02       0.50499998E-02       0.49500000E-02
    7777       0.54600001E-02       0.50250003E-02       0.49500000E-02
    7778       0.54600001E-02       0.49999999E-02       0.49500000E-02
       1       2       1     166     167      13      12     177     178
       2       3       2     167     168      14      13     178     179
       3       4       3     168     169      15      14     179     180
       4       5       4     169     170      16      15     180     181
       5       6       5     170     171      17      16     181     182
       6       7       6     171     172      18      17     182     183
       7       8       7     172     173      19      18     183     184
       8       9       8     173     174      20      19     184     185
       9      10       9     174     175      21      20     185     186
      10      11      10     175     176      22      21     186     187
       .
       .
       .
    5550    7725    7724    7767    7768    7730    7729    7771    7772
    5551    7726    7725    7768    7769    7731    7730    7772    7773
    5552    7727    7726    7769    7770    7732    7731    7773    7774
    5553    7729    7728    1813    7771    7734    7733    1814    7775
    5554    7730    7729    7771    7772    7735    7734    7775    7776
    5555    7731    7730    7772    7773    7736    7735    7776    7777
    5556    7732    7731    7773    7774    7737    7736    7777    7778
    5557    7734    7733    1814    7775    7739    7738    1815    7406
    5558    7735    7734    7775    7776    7740    7739    7406    7401
    5559    7736    7735    7776    7777    7741    7740    7401    7396
    5560    7737    7736    7777    7778    7742    7741    7396    7391
    1    1    0.00000
    2    1    0.00000
    3    1    0.00000
    4    1    0.00000
    5    1    0.00000
    6    1    0.00000
    7    1    0.00000
    8     1    0.00000
    9     1    0.00000
   10     1    0.00000
   11     1    0.00000
    .
    .
    .
    1     2    0.00000
    2     2    0.00000
    3     2    0.00000
    4     2    0.00000
    5     2    0.00000
    6     2    0.00000
    7     2    0.00000
    8     2    0.00000
    9     2    0.00000
   10     2    0.00000
   11     2    0.00000
    .
    .
    .
    1     3   -0.00100
    2     3   -0.00100
    3     3   -0.00100
    4     3   -0.00100
    5     3   -0.00100
    6     3   -0.00100
    7     3   -0.00100
    8     3   -0.00100
    9     3   -0.00100
   10     3   -0.00100
   11     3   -0.00100

---------END OF FILE---------------------------------------------
```

The purpose of this file is to display the format of the input file and is not intended to be comprehensive. The numbers are random and due to practical reasons, a portion of the data has been truncated and has been indicated with '....'.

APPENDIX 6

Sample Output File

3DFEANOF

– U version

3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow – UVWP version

Sample Output File

For a domain discretized into 8-noded prism element

Kee Chien Ting

Advanced Separation Techniques Group
Department of Chemical Engineering
Loughborough University

```
              ************************************************************
              *                                                          *
              *  A  three  dimensional finite element model of a         *
              *  non-newtonian isothermal flow using                     *
              *  the UVP method.                                         *
              *                                                          *
              ************************************************************

            ------------------------------------------------------------------
                        3D Stokes, Sample Output File
            ------------------------------------------------------------------

                    [[[ element description data..........
                        no.of nodes per element                  =        8
                        no.of integration points                 =        3

   *** coordinate system is cartesian (planar) ***

                    [[[ mesh description data ..........
                        no.of nodal points                       =     6560
                        no.of elements                           =     4608
                        no.of nodal constraints on boundary      =     3363
                        no.of different materials                =        1

                    [[[ uniform body force vector ..........
                        grav1                                    =        0.0000
                        grav2                                    =        0.0000
                        grav3                                    =        0.0000

********************************** material properties ************************

  id.     eid.(from-to)   consistency co-efficient     power law index

  1           14608          80.000                        1.3000

 reference temperature   coefficient b   reference pressure  coefficient a

       298.000              0.0000        0.101E+06          0.200

Dispersion Coefficient      Density          Shear rate

   0.200                    0.1E+04           0.20000

******************** nodal coordinates ********************

       id.            x-coord               y-coord               z-coord

        1            0.005100              0.005400              0.005450
        2            0.005137              0.005400              0.005450
        3            0.005175              0.005400              0.005450
        4            0.005213              0.005400              0.005450
        5            0.005250              0.005400              0.005450
        6            0.005288              0.005400              0.005450
        7            0.005325              0.005400              0.005450
        8            0.005363              0.005400              0.005450
        9            0.005400              0.005400              0.005450
       10            0.005100              0.005400              0.005414
        .
        .
        .
     6550            0.005288              0.005050              0.004950
     6551            0.005288              0.005025              0.004950
     6552            0.005288              0.005000              0.004950
     6553            0.005325              0.005075              0.004950
     6554            0.005325              0.005050              0.004950
     6555            0.005325              0.005025              0.004950
     6556            0.005325              0.005000              0.004950
     6557            0.005363              0.005075              0.004950
     6558            0.005363              0.005050              0.004950
     6559            0.005363              0.005025              0.004950
     6560            0.005363              0.005000              0.004950

******************** element connectivity ********************

     id.        n o d a l - p o i n t  e n t r i e s
```

```
        1       2       1     136     137      11      10     145     146
        2       3       2     137     138      12      11     146     147
        3       4       3     138     139      13      12     147     148
        4       5       4     139     140      14      13     148     149
        5       6       5     140     141      15      14     149     150
        6       7       6     141     142      16      15     150     151
        7       8       7     142     143      17      16     151     152
        8       9       8     143     144      18      17     152     153
        9      11      10     145     146      20      19     154     155
       10      12      11     146     147      21      20     155     156
        .
        .
        .
     4600    6517    6516    6551    6552    6522    6521    6555    6556
     4601    6519    6518    1213    6553    6524    6523    1214    6557
     4602    6520    6519    6553    6554    6525    6524    6557    6558
     4603    6521    6520    6554    6555    6526    6525    6558    6559
     4604    6522    6521    6555    6556    6527    6526    6559    6560
     4605    6524    6523    1214    6557    6529    6528    1215    6265
     4606    6525    6524    6557    6558    6530    6529    6265    6260
     4607    6526    6525    6558    6559    6531    6530    6260    6255
     4608    6527    6526    6559    6560    6532    6531    6255    6250

 ******************** nodal constraint ********************

         id.         dof           value

         1           1            0.0000
         2           1            0.0000
         3           1            0.0000
         4           1            0.0000
         5           1            0.0000
         6           1            0.0000
         7           1            0.0000
         8           1            0.0000
         9           1            0.0000
         .
         .
         .
         1           2            0.0000
         2           2            0.0000
         3           2            0.0000
         4           2            0.0000
         5           2            0.0000
         6           2            0.0000
         7           2            0.0000
         8           2            0.0000
         9           2            0.0000
         .
         .
         .
         1           3           -0.0001
         2           3           -0.0001
         3           3           -0.0001
         4           3           -0.0001
         5           3           -0.0001
         6           3           -0.0001
         7           3           -0.0001
         8           3           -0.0001
         9           3           -0.0001

Total number of time steps  =     1

Deltat                      =           0.0100

    iteration no.    1

nodal velocities and  pressures

 id.       ux          uy          uz                   press              stress

   1  0.0000E+00  0.0000E+00  -0.1000E-03        0.91810410E-08       0.11402E-02
   2  0.0000E+00  0.0000E+00  -0.1000E-03        0.82969789E-08       0.12500E-02
   3  0.0000E+00  0.0000E+00  -0.1000E-03        0.79936110E-08       0.11990E-02
   4  0.0000E+00  0.0000E+00  -0.1000E-03        0.80123890E-08       0.12092E-02
   5  0.0000E+00  0.0000E+00  -0.1000E-03        0.91422074E-08       0.12730E-02
   6   0.0000E+00   0.0000E+00  -0.1000E-03         0.79039059E-08      0.11983E-02
   7   0.0000E+00   0.0000E+00  -0.1000E-03         0.78016262E-08      0.11804E-02
   8   0.0000E+00   0.0000E+00  -0.1000E-03         0.80589485E-08      0.12337E-02
   9   0.0000E+00   0.0000E+00  -0.1000E-03         0.93093990E-08      0.16415E-02
  10   0.1345E-03  -0.9495E-04  -0.9798E-04         0.74222439E-08      0.20318E-03
   .
   .
   .
6550  -0.1766E-05   0.1277E-04  -0.1579E-04         0.00000000E+00      0.29442E-04
6551  -0.4432E-05   0.1402E-04  -0.1418E-04         0.00000000E+00      0.42302E-04
6552  -0.7070E-05   0.1553E-04  -0.1465E-04         0.82261104E-09      0.31416E-04
6553  -0.2750E-05   0.1125E-04  -0.1386E-04         0.00000000E+00      0.25755E-04
6554  -0.3105E-05   0.1242E-04  -0.1362E-04         0.00000000E+00      0.31977E-04
6555  -0.5068E-05   0.1347E-04  -0.1190E-04         0.00000000E+00      0.44618E-04
6556  -0.7075E-05   0.1469E-04  -0.1204E-04         0.77467008E-09      0.31905E-04
6557  -0.4796E-05   0.2040E-04  -0.2796E-04         0.00000000E+00      0.28444E-04
6558  -0.4461E-05   0.2292E-04  -0.2534E-04         0.00000000E+00      0.34725E-04
6559  -0.5604E-05   0.2542E-04  -0.2213E-04         0.00000000E+00      0.46929E-04
6560  -0.6836E-05   0.2816E-04  -0.2120E-04         0.76731478E-09      0.32387E-04
node no.        max ux     node no.         min ux
 1982       0.46918106E-03 2946        -0.46918505E-03

node no.        max uy     node no.         min uy
 2280       0.38389884E-03 1274        -0.38390032E-03

node no.        max uz     node no.         min uz
 2098       0.16771381E-03  740        -0.11652409E-03

node no.        max p      node no.         min p
 1224       0.12210963E-07  127         0.00000000E+00


---------END OF FILE---------------------------------------------
```

The purpose of this file is to display the format of the output file and is not intended to be comprehensive. The numbers are random and due to practical reasons, a portion of the data has been truncated and has been indicated with '....'.

3DFEANOF

– P version

3-Dimensional Finite Element Analysis of Non-Newtonian fluid Flow – Continuous Penalty version

Sample Output File

For a domain discretized into 8-noded prism element

Kee Chien Ting

Advanced Separation Techniques Group
Department of Chemical Engineering
Loughborough University

```
              ************************************************************
              *                                                          *
              *  A  three  dimensional finite element model of a         *
              *  non-newtonian isothermal flow using                     *
              *  the Penalty method.                                     *
              *                                                          *
              ************************************************************

            ------------------------------------------------------------------
                 3D Stokes, Sample Output File, Penalty
            ------------------------------------------------------------------

                    [[[ element description data..........
                        no.of nodes per element                  =        8
                        no.of integration points                 =        3

   *** coordinate system is cartesian (planar) ***

                    [[[ mesh description data ..........
                        no.of nodal points                       =     7778
                        no.of elements                           =     5560
                        no.of nodal constraints on boundary      =     3841
                        no.of different materials                =        1

                    [[[ uniform body force vector ..........
                        grav1                                    =        0.0000
                        grav2                                    =        0.0000
                        grav3                                    =        0.0000

************************************ material properties ************************

  id.     eid.(from-to)   consistency co-efficient    power law index

  1           15560          80.000                          1.3000

 reference temperature   coefficient b  reference pressure  coefficient a

       473.000                 0.0000      0.101E+06          0.200

Dispersion Coefficient     Density           Shear rate

   0.200                   0.1E+04            0.20000

Bulk Modulus

    0.10000E+22

******************** nodal coordinates ********************

       id.               x-coord              y-coord              z-coord

        1               0.005100             0.005500             0.005450
        2               0.005140             0.005500             0.005450
        3               0.005180             0.005500             0.005450
        4               0.005220             0.005500             0.005450
        5               0.005260             0.005500             0.005450
        6               0.005300             0.005500             0.005450
        7               0.005340             0.005500             0.005450
        8               0.005380             0.005500             0.005450
        9               0.005420             0.005500             0.005450
       10               0.005460             0.005500             0.005450
        .
        .
        .
     7770               0.005380             0.005000             0.004950
     7771               0.005420             0.005075             0.004950
     7772               0.005420             0.005050             0.004950
     7773               0.005420             0.005025             0.004950
     7774               0.005420             0.005000             0.004950
     7775               0.005460             0.005075             0.004950
     7776               0.005460             0.005050             0.004950
     7777               0.005460             0.005025             0.004950
     7778               0.005460             0.005000             0.004950

******************** element connectivity ********************
```

```
     id.       n o d a l - p o i n t  e n t r i e s

      1      2       1     166     167      13      12     177     178
      2      3       2     167     168      14      13     178     179
      3      4       3     168     169      15      14     179     180
      4      5       4     169     170      16      15     180     181
      5      6       5     170     171      17      16     181     182
      6      7       6     171     172      18      17     182     183
      7      8       7     172     173      19      18     183     184
      8      9       8     173     174      20      19     184     185
      9     10       9     174     175      21      20     185     186
     10     11      10     175     176      22      21     186     187
      .
      .
      .
   5550   7725    7724    7767    7768    7730    7729    7771    7772
   5551   7726    7725    7768    7769    7731    7730    7772    7773
   5552   7727    7726    7769    7770    7732    7731    7773    7774
   5553   7729    7728    1813    7771    7734    7733    1814    7775
   5554   7730    7729    7771    7772    7735    7734    7775    7776
   5555   7731    7730    7772    7773    7736    7735    7776    7777
   5556   7732    7731    7773    7774    7737    7736    7777    7778
   5557   7734    7733    1814    7775    7739    7738    1815    7406
   5558   7735    7734    7775    7776    7740    7739    7406    7401
   5559   7736    7735    7776    7777    7741    7740    7401    7396
   5560   7737    7736    7777    7778    7742    7741    7396    7391

 ******************** nodal constraint ********************

          id.        dof          value

           1          1          0.0000
           2          1          0.0000
           3          1          0.0000
           4          1          0.0000
           5          1          0.0000
           6          1          0.0000
           7          1          0.0000
           8          1          0.0000
           9          1          0.0000
          10          1          0.0000
           .
           .
           .
           1          2          0.0000
           2          2          0.0000
           3          2          0.0000
           4          2          0.0000
           5          2          0.0000
           6          2          0.0000
           7          2          0.0000
           8          2          0.0000
           9          2          0.0000
          10          2          0.0000
           .
           .
           .
           1          3         -0.0001
           2          3         -0.0001
           3          3         -0.0001
           4          3         -0.0001
           5          3         -0.0001
           6          3         -0.0001
           7          3         -0.0001
           8          3         -0.0001
           9          3         -0.0001
          10          3         -0.0001

 Total number of time steps  =    1

 Deltat                      =   0.0100000000

     iteration no.    1

nodal velocities and  pressures
```

```
  id.      ux         uy          uz     press  stress

    1   0.0000E+00   0.0000E+00  -0.1000E-03         0.62958506E+21   -0.62959E+21
    2   0.0000E+00   0.0000E+00  -0.1000E-03         0.42822298E+21   -0.42822E+21
    3   0.0000E+00   0.0000E+00  -0.1000E-03         0.22049034E+21   -0.22049E+21
    4   0.0000E+00   0.0000E+00  -0.1000E-03         0.13135973E+21   -0.13136E+21
    5   0.0000E+00   0.0000E+00  -0.1000E-03         0.40761163E+20   -0.40761E+20
    6   0.0000E+00   0.0000E+00  -0.1000E-03        -0.41242120E+20    0.41242E+20
    7   0.0000E+00   0.0000E+00  -0.1000E-03        -0.88127414E+19    0.88127E+19
    8   0.0000E+00   0.0000E+00  -0.1000E-03         0.72751255E+20   -0.72751E+20
    9   0.0000E+00   0.0000E+00  -0.1000E-03         0.85965128E+20   -0.85965E+20
   10   0.0000E+00   0.0000E+00  -0.1000E-03         0.98436355E+20   -0.98436E+20
    .
    .
    .
 7770  -0.6965E-05   0.8283E-05  -0.1157E-04         0.26301782E+22   -0.26302E+22
 7771  -0.3491E-06   0.1569E-04  -0.2300E-04         0.17516783E+23   -0.17517E+23
 7772  -0.3952E-05   0.1587E-04  -0.1759E-04         0.78990915E+22   -0.78991E+22
 7773  -0.6242E-05   0.1625E-04  -0.1443E-04         0.46274645E+22   -0.46275E+22
 7774  -0.8055E-05   0.1687E-04  -0.1466E-04         0.42187011E+22   -0.42187E+22
 7775  -0.3849E-05   0.1384E-04  -0.3988E-04         0.13214362E+23   -0.13214E+23
 7776  -0.6421E-05   0.1480E-04  -0.3171E-04         0.85554423E+22   -0.85554E+22
 7777  -0.7791E-05   0.1612E-04  -0.2666E-04         0.61066764E+22   -0.61067E+22
 7778  -0.8791E-05   0.1787E-04  -0.2574E-04         0.30416666E+22   -0.30417E+22
node no.         max ux    node no.         min ux
  264        0.33928248E-02 1585       -0.37591549E-02

node no.         max uy    node no.         min uy
  293        0.70772940E-02 1602       -0.17891826E-01

node no.         max uz    node no.         min uz
 3887        0.19925923E-03 5418       -0.20276278E-03

node no.         max p     node no.         min p
 4034        0.26049796E+24 2677       -0.96307804E+23


----------END OF FILE----------------------------------------------------------
```

The purpose of this file is to display the format of the output file and is not intended to be comprehensive. The numbers are random and due to practical reasons, a portion of the data has been truncated and has been indicated with '....'.