Swansea University
Prifysgol Abertawe

Cronfa
Setting Research Free

Swansea University E-Theses

Computational modelling of non-Newtonian fluids based on the stabilised finite element method.

Slijepcevic, Sava

How to cite:

Slijepcevic, Sava (2002) *Computational modelling of non-Newtonian fluids based on the stabilised finite element method..* thesis, Swansea University.
http://cronfa.swan.ac.uk/Record/cronfa42378

Use policy:

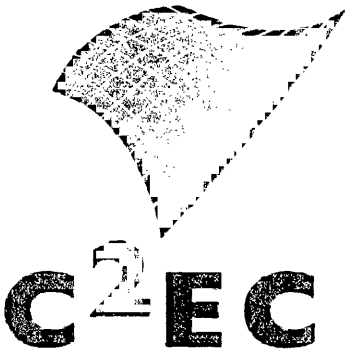
This item is brought to you by Swansea University. Any person downloading material is agreeing to abide by the terms of the repository licence: copies of full text items may be used or reproduced in any format or medium, without prior permission for personal research or study, educational or non-commercial purposes only. The copyright for any work remains with the original author unless otherwise specified. The full-text must not be sold in any format or medium without the formal permission of the copyright holder. Permission for multiple reproductions should be obtained from the original author.

Authors are personally responsible for adhering to copyright and publisher restrictions when uploading content to the repository.

Please link to the metadata record in the Swansea University repository, Cronfa (link given in the citation reference above.)

http://www.swansea.ac.uk/library/researchsupport/ris-support/

CIVIL AND COMPUTATIONAL ENGINEERING CENTRE
SCHOOL OF ENGINEERING
UNIVERSITY OF WALES SWANSEA



COMPUTATIONAL MODELLING OF NON-NEWTONIAN FLUIDS BASED ON THE STABILISED FINITE ELEMENT METHOD

BY

Sava Slijepčević DIPL.ING., MSC

Thesis submitted to the University of Wales
for the award of the degree of PhD

C/Ph/255/02

16/05/2002

ProQuest Number: 10798086

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

ProQuest 10798086

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

UNIVERSITY OF
LIBRARY
WALES SWANSEA

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Candidate: _____

Date: 16/05/2002

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Candidate: _____

Supervisor: _____

Date: 16/05/2002

Statement 2

I hereby consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations

Candidate: _____

Date: 16/05/2002

Acknowledgements

I would like to thank professor Đorđe Perić, my supervisor, for his guidance, support and encouragement during all stages of this work.

I would also like to thank my undergraduate professors and lectures M. Sekulović, Đ. Vuksanović, P. Pujević and R. Salatić at the University of Belgrade, Faculty of Civil Engineering who first introduced me to the finite element method and persuaded me to come to Swansea.

Many thanks are due to everybody in Rockfield Software Ltd that made my work with Elfen© easier and more enjoyable especially A. Garai, G.Q. Liu, A.J.L Crook, J. Armstrong and F. Paw.

I am also grateful to my fellow researchers A. Orlando, D.S. Langley, W. Dettmer, N. Massarotti, A. Rippl and H. Ettinger for many useful discussions and for making my stay in Swansea interesting.

For the unconditional support received, I wish to express my gratitude to my wife Dragana, my parents Erica and Saša and my brother Nikola.

To Dragana, Ognjen,
my parents Erica & Saša
and my brother Nikola

Summary

This work considers numerical modelling of non-Newtonian fluid flow. The motivation for this is the need for simulating industrial processes that involve non-Newtonian fluids.

The non-Newtonian fluids that are included belong to the group of generalised Newtonian fluids. Generalised Newtonian fluids have a non-linear dependence between the shear stain rate and the shear stress, implying a non constant viscosity. Depending on the type of non-linearity generalised Newtonian fluids can be divided into three groups: shear–thinning, shear–thickening and visco–plastic fluids In this work all three groups are considered.

The numerical modelling is performed by using a semi-discrete finite element method. The spatial domain is discretized with finite elements while the time domain is discretized with a discrete time stepping scheme. The finite element method that is used belongs to the group of stabilised finite element methods. Two types of stabilisation are employed. The first is the streamline upwind Petrov-Galerkin method (SUPG) stabilisation that is used to prevent the occurrence of spurious node-to-node oscillations that appear in the presence of dominant advective terms. The second type of stabilisation is pressure stabilisation This stabilisation is necessary to remove pressure oscillations and allow for greater flexibility when choosing interpolation functions. Thus the main unknown variables, velocity and pressure are discretized by linear equal order interpolation functions which offers substantial implementation advantages. The time stepping scheme is a single step method with a generalised midpoint rule. The scheme includes a parameter that can be used to obtain a variety of time stepping schemes from backward Euler to trapezoidal rule. The highly non linear system of equations obtained after discretization is solved via the Newton-Raphson solution procedure.

After the finite element formulation was discretized, consistently linearised and implemented into a finite element program, several tests were performed to verify the implementation, to determine the accuracy and to prove suitability of this type numerical modelling for industrial applications. Several large scale problems from industrial practice have finally been solved to illustrate capabilities of the methodology.

Contents

List of Figures

Chapter 1

Introduction

This work considers the simulation of incompressible non-Newtonian fluid flow. Non-Newtonian fluids are all fluids that have a non-linear relationship between the rate of deformation tensor and the stress tensor. Within this thesis the so-called generalised Newtonian fluids are considered. The source of non-linearity for this type of fluid is related to a shear strain rate dependent viscosity.

The numerical modelling is based on solving the associated initial boundary value problem (IBVP) that describes the fluid flow via the finite element method (FEM). The IBVP consists of a domain of interest, boundary conditions, initial conditions and partial differential equations (PDE) that define the problem. The domain of interest represents space over which the solution is sought. The domain is fixed in space which means an Eulerian coordinate system is used. The boundary conditions represent prescribed values on the domain boundary. The initial conditions correspond to the prescribed values on the entire domain at the initial time step. The PDE's that govern fluid flow are the continuity equation, momentum equation and the non-Newtonian constitutive equations. These equations belong to the group of hyperbolic-parabolic equations. The unknown variables are the velocity and pressure fields. The PDE's relate the velocity and pressure fields in a profound and complex way that makes analytical solution impossible for most real world problems and thus a numerical solution is of great interest.

A number of different non-Newtonian material models are considered. Constitutive models for generalised non-Newtonian fluids can be divided into three groups depending on how the apparent viscosity changes with respect to the shear strain rate. The first group comprise fluids for which the apparent viscosity decreases as the shear strain rate increases and are called shear-thinning fluids. For this group the power law, Bird-Carreau and Cross law fluid models are considered. The second group, the so-called viscoplastic fluids, are characterised by the existence of a yield stress that renders the apparent[1] viscosity infinite until the yield stress is reached. For this group the models that are considered include the Bing-

[1] A useful measure that is used when describing non-Newtonian fluids is the apparent viscosity. It represents the ratio between the shear stress and the shear strain rate.

ham and Herschel-Bulkley models. And finally the third group are shear thickening fluids and comprise fluids for which the apparent viscosity increases as the shear strain rate increases. Such fluid models can be obtained by choosing the appropriate parameters for the power law model (Chhabra, 1993).

The FEM that has been adopted to solve the above mentioned IBVP uses linear interpolation functions to approximate the unknown velocity and pressure fields. The IBVP is converted into a weak form that is then discretized via triangulation of the spatial domain. The triangulation yields a mesh comprising of nodes and elements. The unknown velocity and pressure fields are expressed in terms of values for nodal velocities and nodal pressures. The weak form is weighted at the nodes and then numerically integrated within the elements. This yields a system of ordinary differential equations (ODE) featuring the nodal velocities, nodal pressures and nodal derivatives of the velocities with respect to time (nodal accelerations). The system of ODE is then further converted into a system of algebraic equations by using a time stepping scheme. The system of algebraic equations that is obtained is non-linear and thus requires solution by using an iterative procedure. For this purpose the Newton–Raphson iterative procedure is chosen because of its convergence properties. The Newton-Raphson procedure requires the analytical expression for the tangent stiffness matrix which may present a difficult task. This thesis shows how to obtain such a stiffness matrix for the proposed formulation. The derivation is performed in a methodical way which is useful as a template for future developments.

When the IBVP is converted into a weak form several issues had to be taken into account. First the presence of both velocity and pressure fields makes this a mixed method problem. The choice of interpolation functions for such problems must be approached with care. In particular the interpolation functions used to discretize the original problem must satisfy the so-called Ladyzhenskaya–Babuška–Brezzi (LBB) stability condition (Brezzi, 1974). It is well documented that mixed interpolations that do not satisfy LBB condition lead to instabilities that are manifested as the so-called "checker board" modes. To circumvent the need for the LBB the so-called stabilised methods can be used. We have employed the formulation developed by Hughes and co-workers (Hughes *et al.*, 1986a) and then further developed by Tezduyar and co-workers (Tezduyar *et al.*, 1992).

The second issue that requires consideration is the nature of fluid flow equations. In the case of advection dominated flows these equations belong to the group of hyperbolic-parabolic equations and solving them by using a standard Galerkin weighting in the weak form would result in spurious node-to-node oscillations, so called "wiggles".

To overcome this problem a Petrov-Galerkin scheme has been employed. This scheme modifies the weighting function by adding an optimal amount of artificial diffusion in the direction of the flow. This is a method that was developed by (Brooks & Hughes, 1982) and then further extended by (Tezduyar *et al.*, 1991). This method is termed the streamline upwind Petrov-Galerkin (SUPG). An alterna-

tive to using SUPG for the elimination of the node-to-node oscillations would be to follow the advice of (Gresho & Sani, 1998) and refine the mesh where these oscillations occur. The reason that this method is not used is that the amount of elements that are used need to be minimal since the intention is to use this solution procedure for large scale industrial problems.

For the discretization in time a single step method will be employed. The time stepping scheme is based on evaluating the two factors that make up the convective term at different time instants. The two parameters can be varied to obtain a series of time stepping schemes including the backward Euler and trapezoidal rule (Simo & Armero, 1994).

A number of tests is performed to verify the accuracy and stability of the numerical procedure. First numerical solutions are compared to analytical solutions for simple flows of non-Newtonian fluids. After that further numerical simulations are performed of 2D and 3D problems with more complex geometries. Finally consideration is given to industrial problems related to single and twin extrusion of non-Newtonian fluids.

The motivation for this work was the need to simulate industrial processes. A large area of industrial interest that involve non-Newtonian fluids are extrusion processes. It is for instance, widely used in food processing industries. Therefore the last two tests that where run involve simulation of extrusion.

1.1 Presentation order

This thesis is organised in 7 chapters. Details of the implementation and some additional derivation are included in appendices. The overview of thesis is as follows:

Chapter 1: Introduction
This chapter gives an overview of thesis and provides literature survey.

Chapter 2: Basics of Fluid Mechanics
This chapter gives an overview of fluid mechanics. The aim of this chapter is to establish the mathematical basis that governs non-Newtonian fluid flow. It also introduces the notation that will be used in the latter chapters.

Chapter 3: Non-Newtonian Fluids
This chapter gives an overview of the non-Newtonian fluid models and provides some background information for the behaviour of non-Newtonian fluids. The second part of the chapter gives an analytical solution for the Couette flow for both the power law fluid and the Bingham fluid. These analytical solutions are used later to verify the numerical solution.

Chapter 4: Finite Element Methods
This chapter gives an overview of the finite element method (FEM) in the context of non-Newtonian fluids. It starts by introducing the standard Galerkin

FEM and then describes in detail pressure and streamline upwind stabilisation. This chapter finishes by stating the FEM formulation.

Chapter 5: Solution Procedure
This chapter deals with solution of the FEM formulation given in the previous chapter. The emphasis is on the Newton-Raphson solution procedure and associated linearization procedure.

Chapter 6: Numerical Examples
This chapter includes all the numerical tests. The tests are used to verify numerical accuracy and establish robustness of the scheme.

Chapter 7: Conclusion
This chapter gives conclusions and recommendations for future work.

Appendix Appendices
The appendices describe details of algebra and implementation aspects that were to labourious to be included in the main matter. The appendices are as follows:

A Derivation of $\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h]$
This derivation is part of the evaluation of the stiffness matrix.

B Common Identities
This chapter gives a list of common identities used throughout thesis.

C Using symbolic computation
This chapter illustrates the use of Mathematica©to produce FORTRAN code from the expressions that were obtained in chapter 5.

D Computer Implementation
This chapter deal with details of the implementation of the element routines into the finite element program Elfen.

Chapter 2

Basics of Fluid Mechanics

The purpose of this chapter is to give a brief overview of fluid mechanics and to derive the basic equations. A number of textbooks exist that deal in great detail with the subject of fluid mechanics (Spurk, 1997; Currie, 1993; Chorin & Marsden, 1992; Chung, 1978; Bird *et al.*, 1977, 2001).

Fluid mechanics is concerned with materials that deform without limit when loaded by shearing forces. This means that even with a relatively small shearing force fluids deform infinitely. This is in contrast to *solids* where the *deformation* under a small shearing force is finite. In the case of *fluids* it is the *velocity of deformation* that is finite. As a consequence the velocity serves as one of the variables that is, in a natural way, used in description of the mechanics of fluids.

There are two approaches to derive the basic equations that describe the motion of fluids. The first approach to the problem is from a molecular point of view, which treats the fluid as a cluster of molecules. The governing equations for the motion of fluids are then derived from the molecular motion using the laws of dynamics and probability theory. This method gives good results for light gases, but it has not been developed for liquids or dense gases.

The second approach is to derive the governing equations using the continuum concept. The continuum concept assumes that the fluid consists of continuous matter and the behaviour of individual molecules are ignored. At each point of this matter certain variables that reflect the macroscopic behaviour of fluids are defined. These variables are referred to as "field variables". The field variables that are usually used to describe fluid motion are velocity, pressure, density, temperature and internal energy. These field variables are defined at any point in the continuum and represent an average of the action of molecules that occupy a small volume around this point.

Although the first method is elegant and more consistent it is incomplete for liquids and dense gases. On the other hand the second, continuum approach is valid for all gases and liquids with the restriction that the mean free path is much smaller than the smallest physical length of the flow field that is considered. With liquids the mean free path is one molecule diameter. This restriction is not to severe

for most common engineering problems, since for liquids a square whose side is 1 micrometer contains about 2.5×10^{10} molecules. The only time we would expect the continuum theory to break down is around shock waves or in very thin gases encountered for instance at the edge of the atmosphere. For subsequent derivations the continuum approach will be used.

2.1 Preliminaries

When deriving fluid equations using the continuum approach a reference system has to be chosen. This reference system can be either Eulerian or Lagrangian.

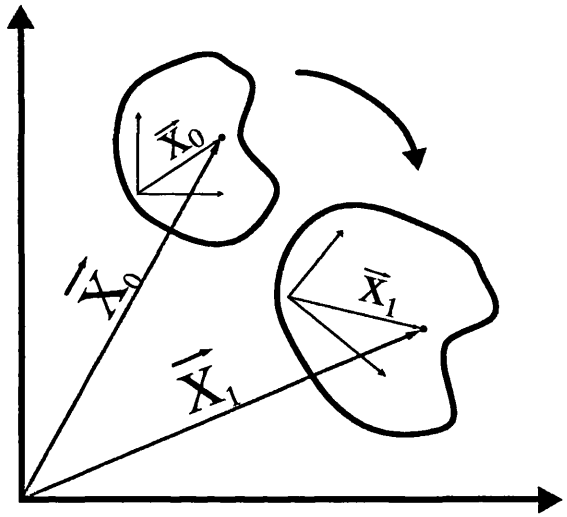


Figure 2.1: Eulerian and Lagrangian reference systems

The Eulerian reference system uses a coordinate system that is fixed in space, with the fluid observations occurring as it passes through a control volume. The control volume is an arbitrary but fixed region in space of volume V and surface S. With this control volume in mind we apply the conservation laws of mass, momentum and energy. The independent variables are the spatial coordinates and time. The Eulerian reference system is also known as the spatial coordinate system.

When using the Lagrangian reference system the coordinate system moves with a particular mass of fluid. The conservation laws are then applied to this particular mass of fluid. The position and volume of the fluid changes with respect to time but there are no fluxes on the surface of the particular mass of fluid which is considered. The Lagrangian reference system is also known as the material coordinate system.

Let x, y, z denote the coordinates of a material point in a fixed coordinate system. The so-called Lagrangian or "material" derivative and the Eulerian derivative are related by the following relationship:

$$\underbrace{\frac{D\alpha}{Dt}}_{Lagrangian} = \underbrace{\frac{\partial\alpha}{\partial t} + u\frac{\partial\alpha}{\partial x} + v\frac{\partial\alpha}{\partial y} + w\frac{\partial\alpha}{\partial z}}_{Eulerian}$$

or in vector form:

$$\frac{D\alpha}{Dt} = \frac{\partial\alpha}{\partial t} + (\mathbf{u}\cdot\nabla)\alpha \tag{2.1}$$

where α is any field variable, $\mathbf{u}$ is the velocity vector and $\frac{D}{Dt}$ is the Lagrangian derivative.

The derivation of the fluid equations involves use of a control volume, which can be either of the fixed or arbitrary size. If a fixed sized control volume is used it is chosen to be vanishingly small and each field variable that is involved in the derivation is expanded in a Taylor series around the centre of the volume. With this arrangement in mind the basic conservation laws are applied. On the other hand if an arbitrary volume that changes its shape and size in time is chosen the conservation principles are applied as an integral over the control volume. This results in a integro-differential equation of the following form:

$$\int_V L\alpha \, dV = 0$$

where L represents some differential operator, V the arbitrary volume and α some property of the fluid. Because V is an arbitrary volume the only way this equation can be satisfied is if $L\alpha = 0$, thus reducing it to a differential equation.

When using an arbitrary control volume and a material (Lagrangian) coordinate system to derive the basic equations, material derivatives of volume integrals are encountered. It is necessary to transform these material derivatives into Eulerian derivatives. This transformation can be done by using the Reynolds' transport theorem.

2.1.1 Proof of the Reynolds' transport theorem

To prove the Reynolds' transport theorem the flow of a specific mass of fluid is considered. In a material framework a fluid property α of a specific mass of fluid is solely dependent on the time t. The starting point in transforming the material derivatives of volume integrals is to use the definition of derivatives:

$$\frac{D}{Dt}\int_{V(t)} \alpha(t)dV = \lim_{\delta t\to 0}\left\{\frac{1}{\delta t}\left[\int_{V(t+\delta t)} \alpha(t+\delta t)dV - \int_{V(t)} \alpha(t)dV\right]\right\}$$

where α is a fluid property and V is the volume. The derivative can be further split into two parts by subtracting and then adding $\alpha(t+\delta t)$ integrated over $V(t)$ as follows

$$\frac{D}{Dt}\int_{V(t)}\alpha(t)dV = \lim_{\delta t\to 0}\left\{\frac{1}{\delta t}\left[\int_{V(t+\delta t)}\alpha(t+\delta t)dV - \int_{V(t)}\alpha(t+\delta t)dV\right] + \frac{1}{\delta t}\left[\int_{V(t)}\alpha(t+\delta t)dV - \int_{V(t)}\alpha(t)dV\right]\right\}$$

The first part of the above equation represents the integration of $\alpha(t+\delta t)$ over the difference in volumes from time t to time $t+\delta t$. The second part of the equation is a familiar Eulerian derivative with respect to time. Now the material derivative can be written as follows:

$$\frac{D}{Dt}\int_{V(t)}\alpha(t)dV = \lim_{\delta t\to 0}\left\{\frac{1}{\delta t}\left[\int_{V(t+\delta t)-V(t)}\alpha(t+\delta t)dV\right]\right\} + \int_{V(t)}\frac{\partial\alpha}{\partial t}dV$$

The integral over the difference in volumes can be transformed into a surface integral over $S(t)$. This can be done by observing that an element of volume δV is equal to $\mathbf{u}\cdot\mathbf{n}\,\delta S$. Here δS is the corresponding surface on $V(t)$ and $\mathbf{u}\cdot\mathbf{n}$ is orthogonal distance between the two volumes $V(t)$ and $V(t+\delta t)$. This yields the following:

$$\frac{D}{Dt}\int_{V(t)}\alpha(t)dV = \lim_{\delta t\to 0}\left\{\frac{1}{\delta t}\left[\int_{S(t)}\alpha(t+\delta t)\mathbf{u}\cdot\mathbf{n}dS\right]\right\} + \int_{V(t)}\frac{\partial\alpha}{\partial t}dV = \int_{S(t)}\alpha(t)\mathbf{u}\cdot\mathbf{n}dS + \int_{V(t)}\frac{\partial\alpha}{\partial t}dV \quad (2.2)$$

The surface integral can be now transformed back into a volume integral by the use of Gauss' theorem:

$$\int_{S(t)}\alpha(t)\mathbf{u}\cdot\mathbf{n}dS = \int_{V(t)}\nabla\cdot(\alpha\mathbf{u})dV \quad (2.3)$$

The final form of the Reynolds' transport theorem is obtained by substituting (2.3) back into (2.2) and is as follows:

$$\frac{D}{Dt}\int_V \alpha\, dV = \int_V\left[\frac{\partial\alpha}{\partial t} + \nabla\cdot(\alpha\mathbf{u})\right]dV \quad (2.4)$$

where α is a scalar field variable and $\mathbf{u}$ is the velocity vector. If α is a vector field variable then the equation can be represented as follows:

$$\frac{D}{Dt}\int_V \boldsymbol{\alpha}\, dV = \int_V\left[\frac{\partial\boldsymbol{\alpha}}{\partial t} + \nabla\cdot(\boldsymbol{\alpha}\otimes\mathbf{u})\right]dV \quad (2.5)$$

where $\boldsymbol{\alpha}$ is a vector field variable. With the above definitions and principles the conservation laws can be derived.

2.2 Conservation of mass

The conservation law for mass states that mass remains unchanged unless chemical reactions take place. This means that an arbitrary mass of fluid can change its shape and size but the mass remains constant. A mathematical way to express this is to state that a material derivative of a mass of fluid contained within an arbitrary volume V is equal to zero. This can written as follows:

$$\frac{D}{Dt}\int_V \rho \, dV = 0$$

The above Lagrangian form of the equation can be transformed into an Eulerian form using the Reynolds' transport theorem (2.4) which gives:

$$\int_V \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u})\right] \, dV = 0$$

By using the fact that the volume is arbitrary implies that the integrand in the above equation has to be equal to zero, which yields the differential equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\text{or } \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + \rho(\nabla \cdot \mathbf{u}) = 0 \tag{2.6}$$

This is the final form of the conservation equation of mass. This equation is often referred to as the *continuity equation* (CE).

If the fluid is incompressible or nearly incompressible the above equation (2.6) can be further reduced using the incompressibility condition. This condition can be mathematically stated as follows:

$$\frac{D\rho}{Dt} = 0 \overset{Using(2.1)}{\Longrightarrow} \frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0 \tag{2.7}$$

When comparing the incompressibility condition (2.7) and the continuation equation (2.6) it is evident that first and second term in the CE are cancelled and the CE for incompressible fluids is reduced to:

$$\boxed{\nabla \cdot \mathbf{u} = 0} \tag{2.8}$$

2.3 Conservation of momentum

This section deals with the derivation of fluid equations that are a consequence of the application of the conservation of momentum to a mass of fluid. Broadly speaking this law states that the change of momentum of a particular fluid mass is equal to the forces acting on it. The momentum per unit volume is equal to $\rho \mathbf{u}$. To get the

total momentum of a particular fluid mass the unit volume momentum has to be integrated over the volume. To get the change of momentum a material derivative of the momentum has to be taken.

Two types of forces are considered. The first type acts on the surface of the fluid mass and is a consequence of the rest of the continuum acting on it. This type of force is manifested through the effect of stresses. To obtain the total force that is acting on the fluid surface, the surface traction vector $\mathbf{t}$ has to be integrated over the surface S. The second type are external body forces that act on the fluid volume and have their source in field forces. They include forces like the gravitational or the electromagnetic forces. They are defined as forces per unit volume. To get the total external body force that is acting on the fluid volume the unit body forces $\mathbf{f}$ have to be integrated over the whole volume V.

The *conservation of momentum* can be represented in the following mathematical form:

$$\frac{D}{Dt}\int_V \rho\mathbf{u}\, dV = \int_S \mathbf{t}\, dS + \int_V \rho\mathbf{f}\, dV \tag{2.9}$$

The value of the surface traction vector $\mathbf{t}$ can be further expressed in terms of stress tensor $\boldsymbol{\sigma}$ as

$$\mathbf{t} = \boldsymbol{\sigma}\mathbf{n} \tag{2.10}$$

where $\mathbf{n}$ is equal to unit normal vector on surface S. After substituting the surface traction vector equation (2.10) and converting the surface integral, using the divergence theorem to a volume integral, the total surface force can be represented as $\int_V \nabla \cdot \boldsymbol{\sigma}\, dV$. The left hand side of equation (2.9) can be transformed into an integral that contains only the Eulerian derivatives using the Reynolds transport theorem. This then yields an equation that consists only of volume integrals and Eulerian derivatives and it can be represented as follows:

$$\int_V \left[\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u})\right] dV = \int_V \nabla \cdot \boldsymbol{\sigma}\, dV + \int_V \rho\mathbf{f}\, dV$$

The fact that V is an arbitrary volume can be used to imply that the integrand on the left and right hand side must be equal and thus yielding the following differential equation:

$$\frac{\partial}{\partial t}(\rho\mathbf{u}) + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = \nabla \cdot \boldsymbol{\sigma} + \rho\mathbf{f}$$

The LHS of the equation can be further simplified by expanding the two terms and using the continuity conditions. If the LHS is expanded by using common

identities (see Appendix B) while observing $\rho\mathbf{u} \otimes \mathbf{u} = \mathbf{u} \otimes \rho\mathbf{u}$ the following is obtained:

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\frac{\partial \rho}{\partial t} + (\rho\mathbf{u} \cdot \nabla)\mathbf{u} + (\nabla \cdot \rho\mathbf{u})\mathbf{u} = \nabla \cdot \boldsymbol{\sigma} + \rho\mathbf{f}$$

Now it can be observed that the second and fourth term on the left hand side represent the continuity equation multiplied by the velocity $\mathbf{u}$ and are therefore equal to zero. The final form is then represented as follows:

$$\boxed{\rho\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot \boldsymbol{\sigma} + \rho\mathbf{f}} \tag{2.11}$$

2.4 Conservation of energy

This section deals with the derivation of fluid equations that are a consequence of the first law of thermodynamics. This law states that the change of the total energy of a body is equal to the sum of the total work done and any heat that has been transferred to the body. The total energy of a body is equal to the sum of the internal energy and the kinetic energy. The internal energy is expressed in terms of internal energy per unit mass e and is obtained by integrating ρe throughout the volume. The kinetic energy is obtained in a analogous fashion by multiplying the unit mass kinetic energy $\frac{1}{2}\mathbf{u}\cdot\mathbf{u}$ with the density ρ an then performing the integration. The total work is split into the work that is done by internal surface forces $\mathbf{u} \cdot \mathbf{t}$ and external body forces $\mathbf{u} \cdot \rho\mathbf{f}$. The heat transferred through the surface of the fluid body can be expressed as a product of the conductive heat flux $\mathbf{q}$ and the surface unit outward normal $\mathbf{n}$, where the heat flux is considered positive when heat is leaving the body. This can be expressed mathematically as follows:

$$\frac{D}{Dt}\int_V \left[\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right] dV = \int_S \mathbf{u}\cdot\mathbf{t}\, dS + \int_V \mathbf{u}\cdot\rho\mathbf{f}\, dV - \int_S \mathbf{q}\cdot\mathbf{n}\, dS$$

As with the previous conservation laws the material derivatives can be transformed into Eulerian derivatives using the Reynolds transport theorem,

$$\int_V \left\{\frac{\partial}{\partial t}\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right) + \nabla\cdot\left[\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right)\mathbf{u}\right]\right\} dV = \int_S \mathbf{u}\cdot\mathbf{t}\, dS + \int_V \mathbf{u}\cdot\rho\mathbf{f}\, dV - \int_S \mathbf{q}\cdot\mathbf{n}\, dS$$

The surface integral $\int_S \mathbf{u}\cdot\mathbf{t}\ dS$ can be transformed into a volume integral by first relating the traction force $\mathbf{t}$ to stress $\boldsymbol{\sigma}$ and then subsequently applying the Gauss' theorem

$$\int_S \mathbf{u}\cdot\mathbf{t}\,dS = \int_S \mathbf{u}\cdot\boldsymbol{\sigma}\,\mathbf{n}\,dS = \int_V \nabla\cdot(\boldsymbol{\sigma}\,\mathbf{u})\,dV$$

The heat transfer surface integral can be also transformed using the Gauss theorem yielding:

$$\int_S \mathbf{q}\cdot\mathbf{n}\,dS = \int_V \nabla\cdot\mathbf{q}\,dV$$

With all these changes applied the energy equation has the form:

$$\int_V \left\{\frac{\partial}{\partial t}\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right) + \nabla\cdot\left[\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right)\mathbf{u}\right]\right\} dV = \int_V \nabla\cdot(\boldsymbol{\sigma}\,\mathbf{u})\,dV + \int_V \mathbf{u}\cdot\rho\mathbf{f}\,dV - \int_V \nabla\cdot\mathbf{q}\,dV$$

Using the arbitrariness of the volume V as in the previous sections a differential equation is obtained:

$$\frac{\partial}{\partial t}\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right) + \nabla\cdot\left[\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right)\mathbf{u}\right] = \nabla\cdot(\boldsymbol{\sigma}\,\mathbf{u}) + \mathbf{u}\cdot\rho\mathbf{f} - \nabla\cdot\mathbf{q}$$

The terms on the LHS contain previously defined conservation equations that can be removed. To achieve this the LHS has to be expanded:

$$\frac{\partial}{\partial t}\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right) = \rho\frac{\partial e}{\partial t} + e\frac{\partial\rho}{\partial t} + \rho\frac{\partial}{\partial t}\left(\frac{1}{2}\mathbf{u}\cdot\mathbf{u}\right) + \frac{1}{2}(\mathbf{u}\cdot\mathbf{u})\frac{\partial\rho}{\partial t} \quad (2.12)$$

$$\nabla\cdot\left[\left(\rho e + \frac{1}{2}\rho\mathbf{u}\cdot\mathbf{u}\right)\mathbf{u}\right] = \nabla\cdot(\rho\mathbf{u}e) + \nabla\cdot\left[\frac{1}{2}(\mathbf{u}\cdot\mathbf{u})\rho\mathbf{u}\right] =$$
$$\rho\mathbf{u}\cdot\nabla e + e\nabla\cdot(\rho\mathbf{u}) + (\rho\mathbf{u}\cdot\nabla)\left(\frac{1}{2}(\mathbf{u}\cdot\mathbf{u})\right) + \frac{1}{2}(\mathbf{u}\cdot\mathbf{u})\,\nabla\cdot(\rho\mathbf{u}) \quad (2.13)$$

It can be noted that the second terms in equation (2.12) and equation (2.13) amount to the continuity equation multiplied by the internal energy e. In a similar way the fourth terms in the above equations represent the continuity equation multiplied by the value of the kinetic energy $\frac{1}{2}(\mathbf{u}\cdot\mathbf{u})$. The third terms in the above equations can also be further simplified and the term $\nabla\cdot(\boldsymbol{\sigma}\,\mathbf{u})$ on the RHS can be expanded. With these simplifications energy equation can now be written as:

$$\rho\frac{\partial e}{\partial t} + \rho\mathbf{u}\cdot\nabla e + \rho\mathbf{u}\cdot\frac{\partial\mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot(\mathbf{u}\cdot\nabla)\mathbf{u} = \mathbf{u}\cdot(\nabla\cdot\boldsymbol{\sigma}) + \boldsymbol{\sigma} : \nabla^{\mathrm{T}}\mathbf{u} + \mathbf{u}\cdot\rho\mathbf{f} - \nabla\cdot\mathbf{q}$$

The above equation can be further reduced by cancelling the terms that amount to the scalar product of the momentum equation and velocity. It has to be noted that $\boldsymbol{\sigma} : \nabla^{\mathrm{T}}\mathbf{u} = \boldsymbol{\sigma} : \nabla\mathbf{u}$ because $\boldsymbol{\sigma}$ is symmetrical. The final form of the energy equation now becomes:

$$\rho\frac{\partial e}{\partial t} + \rho\mathbf{u}\cdot\nabla e = \boldsymbol{\sigma} : \nabla\mathbf{u} - \nabla\cdot\mathbf{q} \tag{2.14}$$

When analysing the physical meaning of (2.14) it can be observed that the rate of change of the internal energy on the LHS is equal to the sum of the transformed mechanical energy and the amount of heat that is added by conduction from outside the body.

2.5 Constitutive equations

To be able to solve the fluid mechanics equations it is necessary to specify two constitutive equations. The first needs to relate the stress tensor $\boldsymbol{\sigma}$ and the deformation rate tensor $\dot{\boldsymbol{\varepsilon}}$ and the second needs to relate the heat-flux gradient $\mathbf{q}$ and the temperature gradients.

2.5.1 Stress - strain rate constitutive equation

The stress that is developed within a moving fluid body is due to deformation. This stress is dependent exclusively on the material properties. To be able to describe material laws the kinematics of a deforming body will first be elaborated.

Kinematics

Kinematics deals with motion and deformation of bodies. To be able to analyse a deformed body it is necessary to define a measure of deformation in the neighbourhood of a point. At a point $\mathbf{x}$ (see Figure 2.2) the velocity is denoted as $\mathbf{u}(\mathbf{x}, t)$ where t is the current time. A neighbouring point $\mathbf{x} + d\mathbf{x}$ has the velocity $\mathbf{u}(\mathbf{x} + d\mathbf{x}, t)$. These two velocities can be related by using the Taylor expansion:

$$\mathbf{u}(\mathbf{x} + d\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, t) + d\mathbf{x}\cdot\nabla\mathbf{u}$$

The velocity gradient $\nabla\mathbf{u}$ can be expanded to the symmetric and the antisymmetric part in the following way:

$$\nabla\mathbf{u} = \frac{1}{2}(\nabla\mathbf{u} + \nabla^{\mathrm{T}}\mathbf{u}) + \frac{1}{2}(\nabla\mathbf{u} - \nabla^{\mathrm{T}}\mathbf{u})$$

The symmetric part is denoted as $\dot{\epsilon}$ and is called the *rate of deformation tensor*,

$$\dot{\epsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla^{\mathrm{T}}\mathbf{u}) = \frac{1}{2}\nabla^{\mathrm{S}}\mathbf{u} = \begin{bmatrix} \frac{du}{dx} & \frac{1}{2}(\frac{du}{dy} + \frac{dv}{dx}) & \frac{1}{2}(\frac{du}{dz} + \frac{dw}{dx}) \\ \frac{1}{2}(\frac{du}{dy} + \frac{dv}{dx}) & \frac{dv}{dy} & \frac{1}{2}(\frac{dv}{dz} + \frac{dw}{dy}) \\ \frac{1}{2}(\frac{du}{dz} + \frac{dw}{dx}) & \frac{1}{2}(\frac{dv}{dz} + \frac{dw}{dy}) & \frac{dw}{dz} \end{bmatrix}$$

The diagonal terms in the *rate of deformation tensor* represent stretching and the non diagonal terms represent shearing deformations respectively. The antisymmetric part is denoted as $\dot{\omega}$ and is called the *spin tensor*,

$$\dot{\omega} = \frac{1}{2}(\nabla \mathbf{u} - \nabla^{\mathrm{T}}\mathbf{u})$$

The value for the velocity in a neighbouring point can now be represented in the following manner:

$$\mathbf{u}(\mathbf{x} + d\mathbf{x}, t) = \mathbf{u}(\mathbf{x}, t) + \dot{\omega}\, d\mathbf{x} + \dot{\epsilon}\, d\mathbf{x}$$

where the terms on the RHS in turn represent the translation, rotation and deformation in the neighbourhood of $\mathbf{x}$. This can be observed in Figure (2.2).

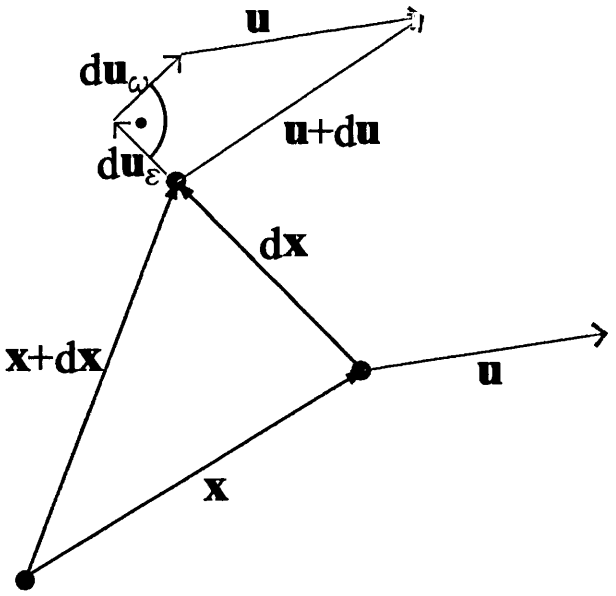


Figure 2.2: Physical significance of the various deformation components

As with all symmetric tensors the *rate of deformation tensor* can be diagonalized. This diagonalization would transform the tensor into a different coordinate

system that has no shearing. This coordinate system is called the principal coordinate system. To obtain the principal coordinate system the following eigenvalue problem needs to be solved:

$$(\dot{\boldsymbol{\epsilon}} - \varepsilon \mathbf{I})\mathbf{w} = \mathbf{0}$$

The non trivial solution of the above equation yields three eigenvalues ε_i, $i = 1, 2, 3$ and three eigenvectors n_i, $i = 1, 2, 3$. The eigenvalues are obtained by solving:

$$\det(\dot{\boldsymbol{\epsilon}} - \varepsilon \mathbf{I}) = 0$$

The above condition represents a cubic equation of the following form:

$$-\varepsilon^3 + \mathrm{I}_1\varepsilon^2 - \mathrm{I}_2\varepsilon + \mathrm{I}_3 = 0$$

where the parameters I_1, I_2 and I_3 are known as the first, second and third invariants of the rate of deformation tensor and are given by the formulae:

$$\mathrm{I}_1 = \mathrm{tr}\,[\dot{\boldsymbol{\epsilon}}]; \quad \mathrm{I}_2 = \frac{1}{2}\left\{(\mathrm{tr}\,[\dot{\boldsymbol{\epsilon}}])^2 - \mathrm{tr}\,[\dot{\boldsymbol{\epsilon}}^2]\right\}; \quad \mathrm{I}_3 = \det(\dot{\boldsymbol{\epsilon}})$$

Since the rate of deformation tensor is symmetric the eigenvalues will be real. Once the eigenvalues are determined the eigenvectors can be calculated. The eigenvectors are orthogonal and represent the principal coordinate system. The eigenvalues represent the deformation along the principal coordinate system. The invariants are unique at a given point. The first parameter represents the volumetric strain. In the case of incompressible fluids it is equal to zero. Also in the case of incompressible fluids the second parameter can be used as a measure of shear strain and is reduced to $-\frac{1}{2}\,\mathrm{tr}\,[\dot{\boldsymbol{\epsilon}}^2]$. A *generalised shear rate* $\dot{\gamma}$ can now be defined as follows:

$$\dot{\gamma} = \sqrt{-4\mathrm{I}_2} = \sqrt{2\,\mathrm{tr}\,[\dot{\boldsymbol{\epsilon}}^2]} = \sqrt{2\,\dot{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}}} \tag{2.15}$$

In the case of a simple shearing flow $\dot{\gamma}$ is obviously equal to the actual shear rate.

Newtonian Fluids

An important group of fluids are the Newtonian fluids. Newtonian fluids are the most abundant fluids on earth, including water and air. For Newtonian fluids the constitutive equation has a simple form in which the stress is related to the rate of deformation tensor in a linear fashion. The relationship has the following form:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \lambda(\nabla \cdot \mathbf{u})\mathbf{I} + 2\mu\dot{\boldsymbol{\epsilon}} \tag{2.16}$$

where μ is the *dynamic viscosity* and λ is the *second viscosity coefficient.* Sometimes instead of using the *dynamic viscosity* the *kinematic viscosity* $\nu = \mu/\rho$ is

used. The *second viscosity coefficient* is a material parameter that is a consequence of the difference between thermodynamic pressure and the mechanical pressure that occurs in polyatomic gases and liquids. This difference is proportional to the divergence of the velocity field $\nabla \cdot \mathbf{u}$. The constant of proportionality is called the *bulk viscosity* $K = \lambda + \frac{2}{3}\mu$. In the case of incompressible fluids the value for λ is not important because the second term is equal to zero due to the continuity equation $\nabla \cdot \mathbf{u} = 0$.

Non-Newtonian Fluids

All fluids that do not have a linear relationship between the rate of deformation tensor and stress fall in the category of non-Newtonian fluids. Non-Newtonian fluids can be further divided into several groups depending on the form of non-linearity. One of the simplest form of non-linear constitutive relation is when the viscosity is not constant and depends on the *generalised shear rate*. The constitutive equation can then be represented in the following way:

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\dot{\boldsymbol{\varepsilon}} \tag{2.17a}$$

$$\mu = \mu(\dot{\gamma}) \tag{2.17b}$$

where $\dot{\gamma}$ represents the *generalised shear rate*. This group of fluids will be discussed in more detail in chapter 3. As with Newtonian fluids for certain compressible fluids the *bulk viscosity* term needs to be added.

2.5.2 Heat flux - temperature gradient constitutive equation

The flux term in the energy equation is a consequence of heat transfer due to conduction. Heat conduction is normally governed by the Fourier's law. Fourier law states that heat conduction is directly proportional to the negative temperature gradient. This law can be mathematically stated as follows:

$$\mathbf{q} = -k\,\nabla T$$

where k is thermal conductivity and represents the proportionality factor.

2.6 Navier - Stokes Equations

The Navier - Stokes equations are obtained by inserting the stress - strain rate constitutive relation for Newtonian fluids (2.16) into the momentum equation (2.11),

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \nabla\cdot(-p\mathbf{I} + \lambda(\nabla\cdot\mathbf{u})\mathbf{I} + 2\mu\dot{\boldsymbol{\varepsilon}}) + \rho\,\mathbf{f}$$

After rearranging the above equation the following is obtained:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla\cdot(\lambda(\nabla\cdot\mathbf{u})) + \nabla\cdot(2\mu\dot{\boldsymbol{\varepsilon}}) + \rho\mathbf{f}$$

This represents the compressible Navier - Stokes equations. For the case of incompressible fluids the following form of the Navier - Stokes equations is used

$$\boxed{\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \nabla^2\mathbf{u} + \rho\mathbf{f}}$$

In the case of negligible viscous effects the second term on the RHS can be dropped and the so-called *Euler equation* are obtained:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \rho\mathbf{f}$$

2.7 Initial Boundary Value Problem (IBVP) for incompressible non-Newtonian fluids

The first step in numerical modelling of non-Newtonian fluid flows involves specifying the initial boundary value problem (IBVP) that describes the problem. An IBVP in general consists of a set of differential equations, a specification of the domain of interest, boundary conditions (BCs) and initial condition (ICs). The differential equations that describe the flow of incompressible non-Newtonian fluids are the momentum equations (2.11) and continuity equation (2.8) coupled with the non-Newtonian constitutive equations (2.17).

The specified domain of interest is an open subset Ω of $\mathbb{R}^n$ where $n = 2, 3$. The boundary of Ω is marked by Γ. The boundary Γ consists of two parts the so called Dirichlet boundary conditions where the prime unknowns are prescribed and the Neumann boundary conditions where the derivatives of the prime unknowns are defined. The Dirichlet boundary is denoted by $\Gamma_{\mathbf{g}}$ and the Neumann boundary is denoted by $\Gamma_{\mathbf{h}}$. The initial condition represents the value of the unknown function all over the domain at some starting time t_0. The domain is shown in Figure 2.3. The IBVP is summarised in the box below.

2.8 Conclusion

In this chapter the basic equations of fluid mechanics have been derived, by following an arbitrary volume of fluid in a material (Lagrangian) coordinate system and applying in turn the basic conservation laws for mass, momentum and energy. During the derivation material derivatives of volume integrals have been transformed

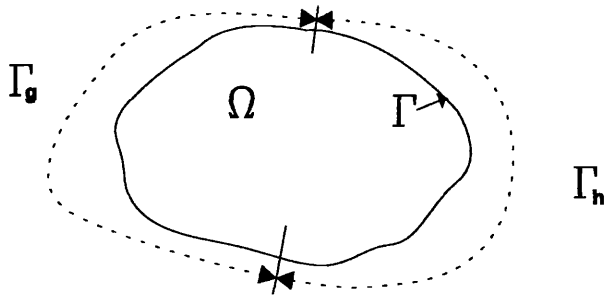


Figure 2.3: Domain and boundary conditions

$$
\begin{aligned}
PDEs: \quad & \rho\tfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} = \nabla\cdot\boldsymbol{\sigma} + \rho\,\mathbf{f} \\
& \boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\dot{\boldsymbol{\epsilon}} \\
& \mu = \mu(\dot{\gamma}); \quad \dot{\gamma} = \sqrt{2\dot{\boldsymbol{\epsilon}} : \dot{\boldsymbol{\epsilon}}} \\
& \dot{\boldsymbol{\epsilon}} = \tfrac{1}{2}\nabla^{\mathrm{S}}\mathbf{u} = \tfrac{1}{2}(\nabla\mathbf{u} + \nabla^{\mathrm{T}}\mathbf{u}) \\
& \nabla\cdot\mathbf{u} = 0 \\
BCs: \quad & \mathbf{u} - \mathbf{g} = 0 \text{ on } \Gamma_{\mathbf{g}} \\
& \boldsymbol{\sigma}\cdot\mathbf{n} - \mathbf{h} = 0 \text{ on } \Gamma_{\mathbf{h}} \\
ICs: \quad & \mathbf{u}(t_0) = \mathbf{u}_0 \quad p(t_0) = p_0
\end{aligned}
$$

Box 2.1

into Eulerian derivatives by using the Reynolds' transport theorem. Since the conservation laws are not sufficient to accurately describe the behaviour of fluid flow additional constitutive laws need to be introduced. Constitutive laws describe the stress - strain rate relationship for Newtonian fluids and the heat flux - temperature gradient relationship. The constitutive laws for non-Newtonian fluids will be discussed in detail in the following chapter. By using the stress - strain rate constitutive law for Newtonian fluids and the momentum equation, the Navier-Stokes equation were derived. Finally an initial boundary value problem for non-Newtonian fluids has been stated which will be the starting point for deriving a stabilised finite element method for non-Newtonian fluids.

Chapter 3

Non-Newtonian fluids

3.1 Introduction to Non-Newtonian Fluids

The most common assumption used in description of fluid behaviour is based on the so-called Newtonian postulate. The Newtonian postulate states that there is a linear relationship between the shear stress (τ) and the shear rate ($\dot{\gamma}$),

$$\tau = \mu\dot{\gamma}$$

where the constant of proportionality μ is called the Newtonian viscosity. This linear relationship is valid for gases, low molecular weight liquids and their solutions but is inadequate to model many real fluids. All those fluids that can not be modelled by using the Newtonian postulate are termed non-Newtonian fluids. Non-Newtonian fluids are encountered in a wide range of industrial and everyday situations such as multi phase mixtures (slurries, emulsions, and gas-liquid dispersions); polymer melts and solutions; soap solutions; cosmetics and toiletries; food products (jams, jellies, cheese, butter, mayonnaise, meat extract, soups yogurt, etc.); biological fluids (blood); natural products (gum, protein solutions, extracts, etc.) and agricultural and diary wastes. This has prompted the need to further develop constitutive models for fluids to accommodate for all these non linearities.

An important class of non-Newtonian fluids are the so-called macro molecular fluids i.e. fluids that consist of large molecules. The molecular weight of macro molecules is in the range $10^4\frac{g}{mol} - 10^9\frac{g}{mol}$. This molecular weight is very high when compared to small molecular fluids for instance water where the molecular weight is equal to $\sim 18\frac{g}{mol}$. The main group of macro molecular fluids are polymers. Polymers consist of large molecules that are composed of many small chemical units, generally called structural units. Polymers can be made out of a chain of structural units or they can be branched. It is important to distinguish between natural (biological) macromolecules which contain a large number of different structural units on one hand and synthetic macromolecules or polymers on the other where the number of different structural units is small. Some polymers have only one structural unit

and they are called homo polymers. Apart from the molecular weight the behaviour of polymers is influenced by their molecular weight distribution. We rarely deal with polymers in which all the molecules have exactly the same weight. In concentrated solutions another source of nonlinearity, is a consequence of temporary entanglement networks. These networks depend on the type of flow and can change with time.

There are various experiments that can be used to show non linearities and show how dramatic the differences between Newtonian and non-Newtonian fluids can be. Typically, the non-Newtonian fluids show shear thinning and shear thickening effects.

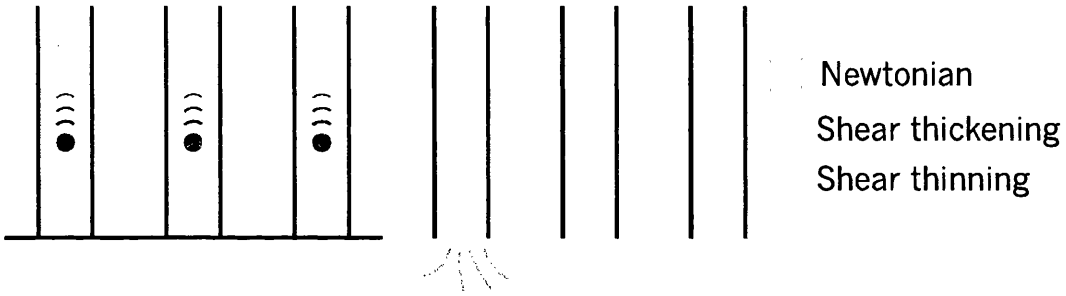


Figure 3.1: Experiment with tube flow

These effects can be demonstrated using a tube flow. A schematic diagram that illustrates an experiment with tube flow can be found in Figure 3.1. Three tubes are filled with a Newtonian fluid, a shear thinning fluid and a shear thickening fluid with the same apparent viscosity. The same apparent viscosity is tested by dropping an identical ball in all fluids and insuring that the velocities are the same. This part of the experiment involves low shear rates. Now once the viscosities have been checked the bottoms of the tubes are removed and the time that the fluids need to drain are recorded. With shear thinning fluids it can be observed that they drain much more quickly than there Newtonian counterpart. With shear thickening the opposite can be observed. A still different behaviour can be observed with viscoplastic fluids. These fluids will not drain at all if the critical stress is not reached. Depending on the diameter of the tube the viscoplastic fluid will not run at all or if the diameter is big enough it will run but in a plug like flow leaving part of the fluid where the critical stress is not reached inside the tube. This experiment is a consequence of a strain rate dependent viscosity.

Apart from strain rate dependent viscosity, there are other sources of nonlinearities such as time dependence and partial elastic recovery. These effects can be demonstrated in experiments such as rod-climbing (Weissenberg effect), axial annular flow, hole pressure error, extrudate swell, secondary flows in the disk and cylinder system, convex surface in a tilted trough, recoil, the tube-less syphon, the Uebler effect (flow through a sudden contraction), drag reduction in turbulent flow and vortex inhibition (Barnes *et al.*, 1989).

Non-Newtonian fluids as demonstrated above have different types of non-linearities. Depending on the type of nonlinearity non-Newtonian fluids can be classified into the following three categories:

(a) Substances for which the viscosity is only dependent on the value of the shear rate. These substances are time independent. They are also known as *generalised Newtonian fluids.*

(b) In a more complicated category are substances for which the viscosity, apart from depending on the value of the shear rate, depends also on the duration of shearing. These substances are termed time dependent.

(c) An even more complicated category of substances that apart from exhibiting fluid properties, also exhibit properties that are characteristic for solids, like partial elastic recovery. These substances are known as viscoelastic fluids.

In the following sections the first category will be discussed in detail, whereas the other two categories will only be briefly mentioned.

3.2 Generalised Newtonian Fluids

Generalised Newtonian fluids are time independent fluids where the only source non-linearity is the viscosity. The viscosity for these types of fluids can be represented in the following form:

$$\mu = \mu(\dot{\gamma})$$

where μ is the viscosity and $\dot{\gamma}$ represents the generalised shear rate and is equal to $\dot{\gamma} = \sqrt{2\mathbf{D} : \mathbf{D}}$. This group of fluids can be further sub-divided into the following three distinct groups:

- Shear thinning or Pseudo-plastic fluids
- Visco plastic fluids
- Shear-thickening or dilatant fluids

The differences between these various types can be best viewed on a shear rate against shear stress graph given in Figure 3.2.

3.2.1 Shear-thinning or Pseudo-plastic Fluids

Pseudo-plastic behaviour is characterised by a varying apparent viscosity that decreases with increasing shear strain rates. Recall that the apparent viscosity can be defined as the shear stress divided by the shear strain rate. The decrease in viscosity often follows a power law dependency. Most non-Newtonian fluids exhibit some

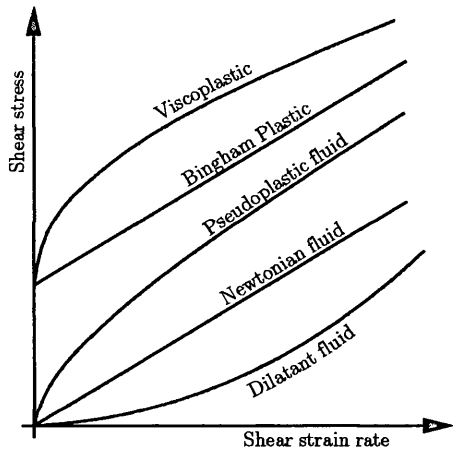


Figure 3.2: Qualitative flow curves

sort of pseudo-plastic behaviour. On the other hand some pseudo-plastic fluids also exhibit regions of constant viscosity. These regions might appear both at very low and at very high shear strain rates. For macro molecular fluids a link exists between the molecular weight distribution and the way in which the viscosity changes from a constant value for low values of strain rate to a power law dependency. Narrow molecular weight distributions give a sharp transition from the constant value region to the power law region. The opposite is true for macro molecular fluids that have broader weight distributions. There is a large number of mathematical models that have been developed for modelling this type of non-Newtonian fluids. The Power Law, Bird-Carreau and Cross Law models represent well known examples.

Power Law Fluid Model The power law is one of most widely and best known fluid models used to describe the behaviour of non-Newtonian fluids. It is based on the fact that most non-Newtonian fluids exhibit a region of linear relationship between stress and strain rate when viewed on a log-log plot. This kind of relationship can be represented in the following way:

$$\tau = m(\dot{\gamma})^n \text{ or } \mu(\dot{\gamma}) = m(\dot{\gamma})^{n-1} \tag{3.1}$$

where m and n are material parameters and represent the *fluid consistency index* and the *power law index* respectively. For shear-thinning fluids $n < 1$. Some typical values for m and n and their corresponding range of validity are given in Table 3.1. The range of validity is given in terms of shear strain rate.

Although the power law model is widely used it has a serious limitation since it only models a specific region of interest. Outside that region the model is not valid. This limitation does not have to be important if the range of interest falls within the range of validity. The areas that are not described with the power law model are those near $\dot{\gamma} = 0$ and when $\dot{\gamma} \to \infty$. This limitation becomes a problem

Liquid	Temp. (K)	n (-)	m $(Pa \cdot s^n)$	Shear rate range (s^{-1})
0.77% Aqueous carboxymethyl cellulose solution	294	0.95	0.044	44-560
0.10% Aqueous Sepran© MG-500 solution	294	0.55	0.205	0.56-883
32.0% Aqueous kaolin suspension	303	0.103	19.5	1-50

Table 3.1: Typical values of m and n (Chhabra, 1993)

Liquid	η_0 $(Pa\dot{s})$	η_∞ $(Pa\dot{s})$	λ (s)	n (-)
2% PIB in Primol 355	923.0	0.15	191.00	0.36
7% Aluminium soap in decalin/m-cresol	86.6	0.01	1.41	0.20
High density polyethylene	8920.0	0.00	1.58	0.50

Table 3.2: Typical values for η_0, η_∞, λ and n (Chhabra, 1993)

when this model is used for numerical simulation as a wide range of values for shear strain rate are used. To solve this problem various modification to the original function need to be used and additional stress-strain relationships need to be used outside the range of validity.

Bird-Carreau Fluid Model This model represents an a extension of the power law model to a four parameter fluid model. Its form is similar to the power law model but it takes into account the values for zero and infinite viscosities. Unlike the power law the Bird-Carreau fluid model is valid for a wider range of shear strain rate $\dot{\gamma}$. The value for viscosity has the following form:

$$\mu(\dot{\gamma}) = \eta_\infty + (\eta_0 - \eta_\infty)\left[1 + (\lambda\dot{\gamma})^2\right]^{\frac{n-1}{2}}$$

where η_0 and η_∞ are the zero and infinite viscosities respectively, λ is a time constant ($1/\lambda$ is equal to the value of shear rate where the flow curve departs from the constant viscosity region) and n is the power law index. The power law index n has the same meaning as for the power law model. Some typical values for the parameters can be found in Table 3.2.

Cross Law Fluid Model Another fluid model for shear-thinning fluids is the Cross law fluid model. This model is very similar to the Bird-Carreau fluid

model. It is used when the value of η_∞ is very small and is negligible. The viscosity formula for this model has the following form:

$$\mu(\dot{\gamma}) = \frac{\eta_0}{1 + (\lambda\dot{\gamma})^m}$$

The parameters have the same meaning as for the Bird-Carreau fluid model, where $m = \frac{n-1}{2}$.

3.2.2 Visco plastic Fluids

Visco plastic behaviour is characterised by the existence of a yield stress. If the yield stress is not exceeded the value for the viscosity is infinite. The yield stress is somewhat of a contentious issue; on one hand we have the fact that fluids do flow below the yield stress in a creep like manner (Barnes, 1999; Barnes & Walters, 1985) so that the yield stress does not really exist, and on the other hand, in practical situations, the concept of yield stress has proved to be very useful. The yield stress can be measured using a variety of tests such as the slump test (Pashias *et al.*, 1996). A diagram that illustrates the slump test can be found in Figure 3.3.

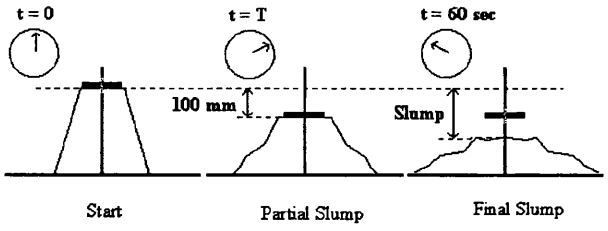


Figure 3.3: Slump test

Once the fluid starts to flow the shear stress - shear strain rate relationship can either be linear in which case we have a Bingham fluid, or nonlinear. The nonlinear part can be modelled with any of the models that are used for pseudo-plastic behaviour. If the power law model is used, then the Herschel-Bulkley model is obtained.

Bingham Fluid Model The Bingham fluid model is one of the best known viscoplastic fluid models. It was developed at the beginning of the century by E.C. Bingham while he was working on clay suspensions. The fluid model has a very simple form that includes parameters,

$$\begin{aligned} \sigma &= \sigma_Y + K\dot{\gamma} \quad \sigma \geq \sigma_Y \\ \mu(\dot{\gamma}) &\to \infty \qquad\quad \sigma < \sigma_Y \end{aligned} \tag{3.2}$$

where σ_Y is the yield stress and K is the plastic viscosity.

When using the Bingham model to simulate non-Newtonian fluid flow the discontinuity at $\dot{\gamma} = 0$ can create numerical difficulties. To overcome this, instead of using the original formula it has proved convenient to use the Papanastasiou's modification (Papanastasiou, 1987). This modification includes a exponential term that smoothes the flow curve,

$$\mu(\dot{\gamma}) = \frac{\sigma_Y}{\dot{\gamma}} \left[1 - e^{-m\dot{\gamma}}\right] + \mu_0$$

where m is the stress growth exponent. The effect of the stress growth exponent on the flow curve can be observed in Figure 3.4.

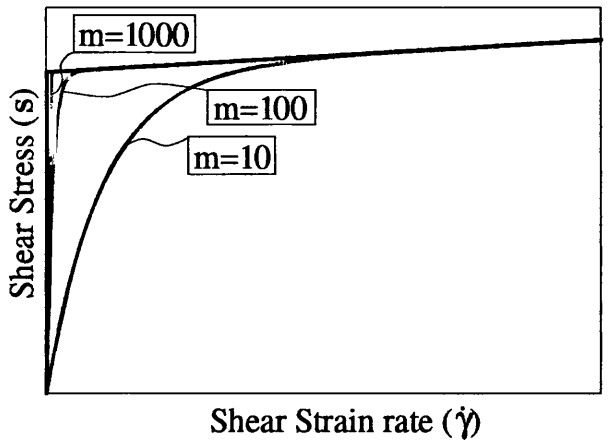


Figure 3.4: A Bingham fluid flow curve using the Papanastasiou's modification for various values of the stress growth exponent m.

The Bingham fluid model has proved very useful in describing the behaviour of suspensions. The parameters can be related empirically (Thomas, 1961) to the viscosity of the liquid μ_s, the diameter of the particles D_p and the concentration ϕ. This empirical relationship has the following form:

$$\sigma_Y = 312.5 \frac{\phi^3}{D_p^2}$$

$$K = \mu_s \exp\left[\phi\left(\frac{5}{2} + \frac{14}{\sqrt{D_p}}\right)\right]$$

Parameters for the Bingham fluid model can also be obtained experimentally. An example of an experimental flow curve with the determined parameters is shown in Figure 3.5.

Herschel-Bulkley Fluid Model When combining the Bingham fluid model and the power law model, the so-called Herschel-Bulkley fluid model is obtained (Chhabra, 1993). This model can be described in the following way:

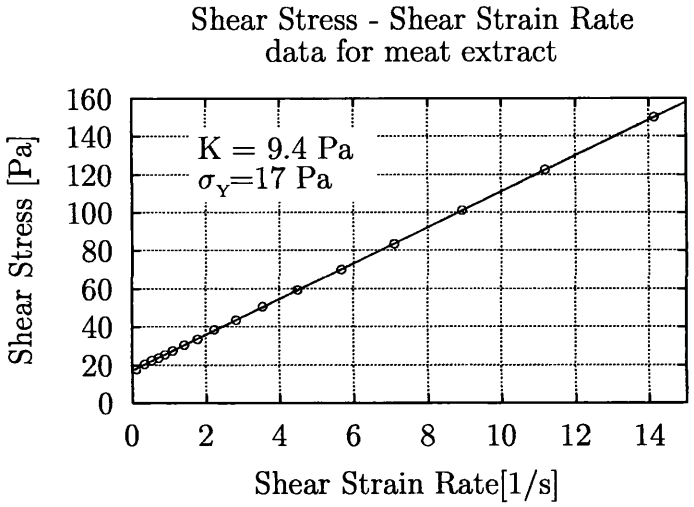


Figure 3.5: Experimental data for a meat extract (T=350K) that has a Bingham plastic behaviour (Reproduced from (Chhabra, 1993)).

$$\sigma = \sigma_Y + K\dot{\gamma}^n \qquad \sigma \geq \sigma_Y$$
$$\mu(\dot{\gamma}) \to \infty \qquad \sigma < \sigma_Y$$

where the parameters σ_Y, K and n have the same meaning as for the Bingham and power law model.

As with the Bingham model the Herschel-Bulkley model needs to be regularised for low values of strain rate. This can be done by using the Papanastasiou's idea, which yields the following:

$$\mu(\dot{\gamma}) = \frac{\sigma_Y}{\dot{\gamma}}\left[1 - e^{-m\dot{\gamma}}\right] + K\dot{\gamma}^{n-1}$$

3.2.3 Shear thickening Fluids

Shear thickening or dilatant fluids as they are also known, are very similar to shear thinning fluids with the difference that the apparent viscosity increases with an increase in shear strain rate. This kind of behaviour is not encountered as often as shear thinning. It is mostly associated with concentrated suspensions where the liquid phase is just sufficient to fill the voids between the suspended particles. The increase in the apparent viscosity for concentrated suspensions at higher shear rates is caused by the increase in solid-solid friction that is a consequence of the lack of liquid to lubricate the moving particles. The lack of liquid is caused by an increase in voids as the fluid expands or dilates to accommodate for higher strain rates. Not

all shear thickening fluids have an increase in the apparent viscosity as a consequence of the dilatant mechanism, never the less they are also called dilatant fluids. This increase of the apparent viscosity can be described by using a power law model where the power law index is taken as $n > 1$.

3.3 Time Dependent Fluids

Time dependent fluids change their apparent viscosity with time. This type of behaviour can be found in polymeric fluids. They consist of molecules with very high molecular weight. The molecules that make up polymeric fluids can assume a large number of configurations. During flow the configurations are altered by the stretching, alignment and entanglement of molecules. The various configurations are possible because parts of the molecules can rotate around chemical bonds. This behaviour is governed by Brownian motion forces, intermolecular forces and hydrodynamic forces. As a result their apparent viscosity changes over time. If the apparent viscosities decreases over time these fluids are said to exhibit Thixotropy. Rheopexy is used to term the opposite behaviour. More details about polymeric fluids can be found in Bird *et al.* (1977).

Thixotropy With these materials the apparent viscosity decreases with time and is a consequence of the breakdown of internal structural links and alignment of molecules. These links are a consequence of intermolecular forces. As the number of broken structural links increases the number of links that are available to reform increases. This process eventually reaches a dynamic equilibrium and the rate of change of viscosity with respect to time goes to zero.

Rheopexy There are a few fluids that exhibit Rheopexy. This kind of behaviour is a consequence of a gradual build up of the internal structure. The structure is the result of molecules creating temporary entanglement networks.

3.4 Viscoelastic Fluids

Fluids that show partial elastic recovery are called viscoelastic fluids. Elastic behaviour implies that the normal stresses are not equal in general. When analysing viscoelastic flows it is useful to define the axis as follows: the first axis is in the flow direction, the second is in the direction of the flow gradient and the third is orthogonal to the first two. With these axis in mind the primary normal stress difference is defined as the difference of the normal stresses of the first two axis. The secondary normal stress difference is calculated by using the normal stresses along the second two axis. These stress differences can be used to differentiate between viscoelastic

fluids and non-viscoelastic fluids. For viscoelastic fluids the primary and secondary normal stress differences are not equal to zero.

More details about time dependent fluids and viscoelastic fluids can be found in Bird *et al.* (1977); Tanner (1992); Harris (1977); Barnes *et al.* (1989).

3.5 Remarks on computational modelling of non-Newtonian fluids

When developing computational models it is desirable that constitutive relationships have a degree of continuity and that they cover the entire spectrum of possible values for parameters and independent variables. Most of the models that have been developed to model non-Newtonian fluids have been developed to fit a narrow set of experimental data. This comes from the fact that they have been developed for certain engineering and industrial purposes. An additional source of problem is that during the solution procedure values for the independent variables that do not occur in nature may be encountered. All these issues often require that the models need to be modified and extended. The Papanastasiou's modification to the Bingham equation is an example of such a modification which enforces C^1 continuity for $\dot{\gamma} = 0$. Another example of a constitutive model that needs to be modified is the power law model. The problem with this model is exhibited when the shear strain rate is close to zero or in limit case equal to zero, the apparent viscosity approaches infinity. The problem may be solved by defining two forms of the relationship between shear strain rate and stress. One for use below a critical value of the shear strain rate and the other is the standard form of the power law model for use above the critical shear strain rate. The alternative form is a linear function that is determined by imposing C^0 and C^1 continuity at the critical shear strain rate.

The regions of shear strain rate $\dot{\gamma}$ that need to be considered when modelling non-Newtonian fluids are $\dot{\gamma} = 0$, $\dot{\gamma} \to 0$ and $\dot{\gamma} \to \infty$. Further details of how these problems can be solved will be elaborated in chapter 5.

3.6 Analytical verification for a simple Couette flow

When making numerical simulations it is necessary to be able to verify the results. One way of doing this is to compare numerical results with analytical solutions. To obtain analytical solutions the basic set of differential equations that were derived in chapter 2 have to be coupled to a constitutive material model, applied to a problem and then solved. This process is lengthy and tedious as will be illustrated in the following section. Because of the nature of the problem analytical solutions are only available for relatively simple problems. As an illustration of the analytical solution procedure a Couette flow is solved in the following.

3.6.1 Differential equation for a Couette flow

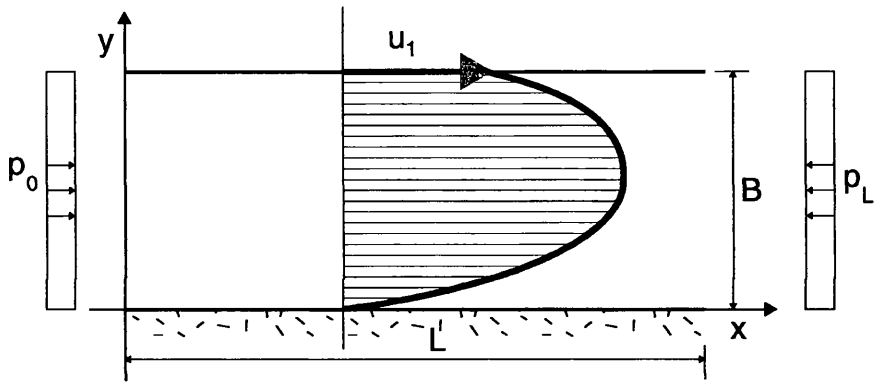


Figure 3.6: Couette flow.

A Couette flow is a flow between two parallel plates where one plate is fixed and the other is moving at a constant speed. A pressure gradient along the flow direction is also present. The solution described in the following is based on Flumerfelt *et al.* (1969). It is extended for Bingham fluids and is adapted for use within a computer program.

To derive an analytical solution for a Couette flow, the basic equations for fluid mechanics from chapter 2 have to be applied in a 2D context. The equations that are needed are the momentum equation (2.11) and a constitutive model for generalised non-Newtonian fluids (2.17). These two equations can be written in a 2D context for a particular Couette flow conditions in the following way:

1 - Momentum equation

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\nabla\mathbf{u}\right) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{0}$$

where the various terms have the following meaning:

- ρ is the density and $\boldsymbol{\sigma}$ is the stress tensor
- $\mathbf{u}$ is the velocity vector

$$\mathbf{u} = \begin{bmatrix} u \\ v \end{bmatrix} \overset{\text{C.F.}}{=} \begin{bmatrix} u \\ 0 \end{bmatrix}$$

- Velocity gradient

$$\nabla\mathbf{u} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{bmatrix} \overset{\text{C.F.}}{=} \begin{bmatrix} 0 & \frac{\partial u}{\partial y} \\ 0 & 0 \end{bmatrix}$$

- The convective term

$$v = 0 \Longrightarrow \mathbf{u}\nabla\mathbf{u} = \begin{bmatrix} u \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & \frac{\partial u}{\partial y} \\ 0 & 0 \end{bmatrix} = \mathbf{0}$$

- Transient term

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{0} \qquad \text{since we are dealing with a steady-state problem}$$

2 - Constitutive model for generalised non-Newtonian fluids

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu(\dot{\gamma})\dot{\boldsymbol{\epsilon}}$$

where

- p is pressure, $\mathbf{I}$ identity tensor and $\mu(\dot{\gamma})$ is the shear rate dependent viscosity
- Strain rate tensor

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla^{\mathrm{T}}\mathbf{u}\right) = \frac{1}{2}\begin{bmatrix} 0 & \frac{\partial u}{\partial y} \\ \frac{\partial u}{\partial y} & 0 \end{bmatrix}$$

The stress tensor in 2D has the following form:

$$\boldsymbol{\sigma} = \begin{bmatrix} -p & \mu(\dot{\gamma})\frac{\partial u}{\partial y} \\ \mu(\dot{\gamma})\frac{\partial u}{\partial y} & -p \end{bmatrix}$$

The diffusion term $\nabla \cdot \boldsymbol{\sigma}$ from the momentum equation has the following form

$$\nabla \cdot \boldsymbol{\sigma} = \begin{bmatrix} -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left(\mu(\dot{\gamma})\frac{\partial \mathbf{u}}{\partial y}\right) \\ \frac{\partial}{\partial x}\left(\mu(\dot{\gamma})\frac{\partial \mathbf{u}}{\partial y}\right) - \frac{\partial p}{\partial y} \end{bmatrix} \overset{\text{C.F.}}{=} \begin{bmatrix} -\frac{\partial p}{\partial x} + \frac{\partial}{\partial y}\left(\mu(\dot{\gamma})\frac{\partial u}{\partial y}\right) \\ 0 \end{bmatrix}$$

By back substituting the diffusion term back into the momentum equation the following second order differential equation that governs a Couette flow is obtained:

$$\boxed{-\frac{\mathrm{d}p}{\mathrm{d}x} + \frac{\mathrm{d}}{\mathrm{d}y}\left(\mu\frac{\mathrm{d}u}{\mathrm{d}y}\right) = 0} \tag{3.3}$$

Equation (3.3) can be further simplified to a first order differential equation irrespective of the value of the viscosity function $\mu(\dot{\gamma})$ by integrating in the X and Y directions (see Figure 3.6). The integration constants are eliminated by applying the relevant boundary conditions (BC).

- Integration in the X direction

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y}\right) = \frac{\mathrm{d}p}{\mathrm{d}x}$$
$$\int \frac{\mathrm{d}}{\mathrm{d}y}\left(\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y}\right)\,\mathrm{d}x = \int \frac{\mathrm{d}p}{\mathrm{d}x}\,\mathrm{d}x + C_1$$
$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y}\right)x = p + C_1$$

- BC in the X direction

$$\text{BC: } p = \begin{cases} p_0 & x = 0 \Rightarrow C_1 = -p_0 \\ p_L & x = L \end{cases}$$

- Integration in the Y direction

$$\frac{\mathrm{d}}{\mathrm{d}y}\left(\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y}\right)L = p_L - p_0$$
$$\int \frac{\mathrm{d}}{\mathrm{d}y}\left(\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y}\right)L\,\mathrm{d}y = \int p_L - p_0\,\mathrm{d}y + C_2$$
$$\left(\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y}\right)L = (p_0 - p_L)y + C_2$$

- BC in the Y direction

$$\text{BC: } u = \begin{cases} u_1 & y = B \\ 0 & y = 0 \end{cases}$$

By substituting $y = B\eta$ where $\eta = [0, 1]$ and $C_2 = \frac{p_0 - p_L}{L}\lambda$, a new unknown λ is obtained. This is a common starting point for an analytical solution for all generalised non-Newtonian fluids:

$$\boxed{\mu(\dot{\gamma})\frac{\mathrm{d}u}{\mathrm{d}y} = \left[\frac{(p_0 - p_L)B}{L}\right](\lambda - \eta)} \tag{3.4}$$

The next step in obtaining an analytical solution for a Couette flow is to substitute a value for the viscosity function $\mu(\dot{\gamma})$ in equation (3.4) and then perform integration. The viscosity function depends on the value of the strain rate. For a Couette flow the strain rate has the following form:

$$\dot{\gamma} = (2\boldsymbol{\epsilon} : \boldsymbol{\epsilon})^{1/2} = \left(2\frac{1}{2}\begin{bmatrix} 0 & \frac{\mathrm{d}u}{\mathrm{d}y} \\ \frac{\mathrm{d}u}{\mathrm{d}y} & 0 \end{bmatrix} : \frac{1}{2}\begin{bmatrix} 0 & \frac{\mathrm{d}u}{\mathrm{d}y} \\ \frac{\mathrm{d}u}{\mathrm{d}y} & 0 \end{bmatrix}\right)^{1/2} = \left|\frac{\mathrm{d}u}{\mathrm{d}y}\right|$$

From the possible models that have been described in section 3.2 the Bingham fluid model and the power law fluid model have been chosen to further illustrate the analytical procedure. First a detailed overview for obtaining an analytical solution for a Bingham fluid model is shown. After that the results for the power law fluid are shown.

3.6.2 Analytical solution for a Bingham fluid

A graph of the problem with a velocity profile for a Bingham fluid can be observed in Figure 3.7.

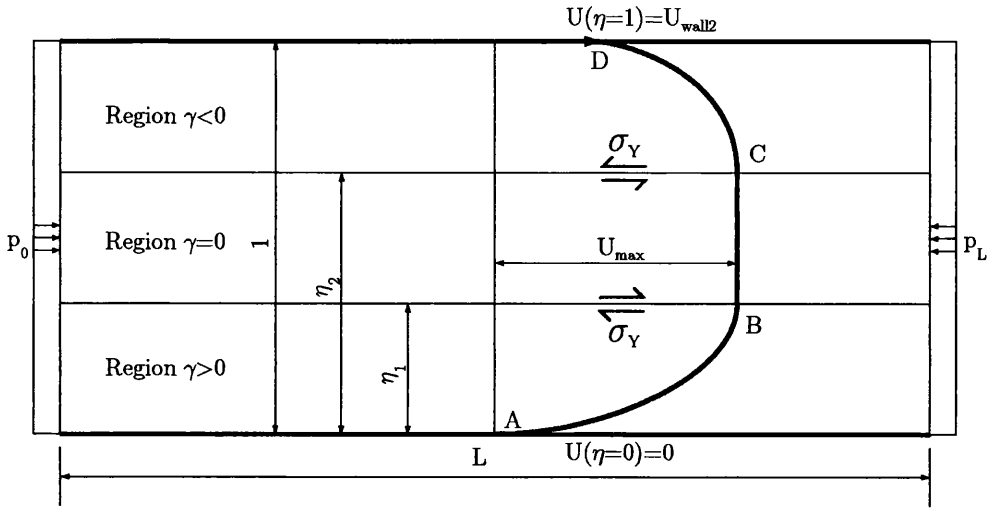


Figure 3.7: Velocity Profile for a Bingham fluid

When the Bingham fluid constitutive model (3.2) is substituted into equation (3.4) the following is obtained:

$$\left(\frac{\sigma_Y}{\left|\frac{du}{dy}\right|} + K\right)\frac{du}{dy} = \frac{(p_0 - p_L)B}{L}(\lambda - \eta) \tag{3.5}$$

Because of the term $\left|\frac{du}{dy}\right|$, the solution of this differential equation has the following three cases:

CASE I, $\frac{du}{dy} > 0$ This is valid from point A to point B

$$\sigma_Y + K\frac{du}{dy} = \frac{(p_0 - p_L)B}{L}(\lambda - \eta)$$

$$\frac{du}{d\eta} = \frac{(p_0 - p_L)B^2}{KL}(\lambda - \eta) - \frac{\sigma_Y B}{K}$$

$$\boxed{u = \frac{(p_0 - p_L)B^2}{KL}\left(\lambda\eta - \frac{\eta^2}{2}\right) - \frac{\sigma_Y B}{K}\eta + C_3} \tag{3.6}$$

CASE II, $\frac{du}{dy} = 0$ This is valid from point B to point C

$$\boxed{u = u_{max}, \tau_{xy} = \sigma_Y, \dot{\gamma} = 0} \tag{3.7}$$

CASE III, $\frac{du}{dy} < 0$ This is valid from point C to point D

$$\frac{du}{d\eta} = \frac{(p_0 - p_L)B^2}{KL}(\lambda - \eta) + \frac{\sigma_Y B}{K}$$

$$\boxed{u = \frac{(p_0 - p_L)B^2}{KL}(\lambda\eta - \frac{\eta^2}{2}) + \frac{\sigma_Y B}{K}\eta + C_3} \tag{3.8}$$

Determining boundary conditions

The boundary conditions are derived from the following conditions:

- Prescribed velocities at the wall points A and D
- C^0 and C^1 continuity at points B and C
- Equilibrium condition between the external forces acting on the region of constant velocity part of the flow (B-C) and shear forces at point B and C

Point D: $u_2(1) = u_{wall2}$

$$u_{wall2} = u_2(1) = \frac{(p_0 - p_L)B^2}{KL}(\lambda_2 - 0.5) + \frac{\sigma_Y B}{K} + C_3^C \tag{3.9}$$

Point C: $u_2(\eta_2) = u_{max} \qquad \frac{du_2}{d\eta} = 0$

$$u_{max} = u_2(\eta_2) = \frac{(p_0 - p_L)B^2}{KL}\left(\lambda_2\eta_2 - \frac{\eta_2^2}{2}\right) + \frac{\sigma_Y B}{K}\eta_2 + C_3^C \tag{3.10}$$

$$\frac{du_2}{d\eta_2} = 0 = \frac{(p_0 - p_L)B^2}{KL}(\lambda_2 - \eta_2) + \frac{\sigma_Y B}{K} \tag{3.11}$$

Point B: $u_1(\eta_1) = u_{max} \qquad \frac{\mathrm{d}u_1}{\mathrm{d}\eta} = 0$

$$u_{max} = u_1(\eta_1) = \frac{(p_0 - p_L)B^2}{KL}\left(\lambda_1\eta_1 - \frac{\eta_2^2}{2}\right) - \frac{\sigma_Y B}{K}\eta_1 + C_3^{\mathrm{B}} \tag{3.12}$$

$$\frac{\mathrm{d}u_2}{\mathrm{d}\eta_2} = 0 = \frac{(p_0 - p_L)B^2}{KL}(\lambda_1 - \eta_1) - \frac{\sigma_Y B}{K} \tag{3.13}$$

Point A: $u_1(0) = 0$

$$0 = \frac{(p_0 - p_L)B^2}{KL}\left(\lambda_1 0 - \frac{0^2}{2}\right) - \frac{\sigma_Y B}{K}0 + C_3^{\mathrm{B}} \Rightarrow C_3^{\mathrm{B}} = 0 \tag{3.14}$$

Equilibrium condition in the constant velocity region $u = u_{max}$

$$\eta_2 - \eta_1 = \frac{2\sigma_Y}{Bp_{GR}} \text{ where } p_{GR} = \frac{p_0 - p_L}{L} \tag{3.15}$$

From the above boundary conditions (3.9)-(3.15) the following unknowns can be determined: η_1, η_2, λ_1, λ_2, C_3^{C} and u_{max} using the known values of u_{wall2}, B, σ_Y, p_0, p_L, K and L.

Determining unknown constants

One of the procedures for determining unknown constants from the above system of equations looks like this:

1 - From (3.9) eliminate C_3^{C}

$$C_3^{\mathrm{C}} = u_{wall2} - \frac{(p_0 - p_L)B^2}{KL}(\lambda_2 - 0.5) - \frac{\sigma_Y B}{K} \tag{3.16}$$

2 - From (3.12) eliminate u_{max}

$$u_{max} = \frac{(p_0 - p_L)B^2}{KL}\left(\lambda_1\eta_1 - \frac{\eta_2^2}{2}\right) - \frac{B\sigma_Y}{K}\eta_1 \tag{3.17}$$

3 - From (3.13) & (3.11) get expressions for λ_1 and λ_2

$$\lambda_1 = \eta_1 + \frac{\frac{B\sigma_Y}{K}}{\frac{(p_0-p_L)B^2}{KL}} \qquad \lambda_2 = \eta_2 - \frac{\frac{B\sigma_Y}{K}}{\frac{(p_0-p_L)B^2}{KL}} \tag{3.18}$$

4 - From (3.15) relationship between η_1 and η_2

$$\eta_2 = \eta_1 + 2\frac{\sigma_Y L}{B(p_0 - p_L)} \tag{3.19}$$

5 - From (3.12) and (3.10) eliminate C_3^C

$$\frac{(p_0-p_L)B^2}{KL}(\lambda_1\eta_1-0.5\eta_1^2)-\frac{B\sigma_Y}{K}\eta_1=$$
$$\frac{(p_0-p_L)B^2}{KL}(\lambda_2\eta_2-0.5\eta_2^2)+\frac{B\sigma_Y}{K}\eta_2+C_3^C$$
$$\frac{(p_0-p_L)B^2}{KL}[\lambda_1\eta_1-\lambda_2\eta_2+0.5(\eta_2^2-\eta_1^2)]-\frac{B\sigma_Y}{K}(\eta_1+\eta_2)=C_3^C \quad (3.20)$$

6 - Using (3.16) for C_3^C the following is obtained:

$$\frac{(p_0-p_L)B^2}{KL}[\lambda_1\eta_1-\lambda_2\eta_2+0.5(\eta_2^2-\eta_1^2)]-\frac{B\sigma_Y}{K}(\eta_1+\eta_2)$$
$$=u_{wall2}-\frac{(p_0-p_L)B^2}{KL}(\lambda_2-0.5)-\frac{B\sigma_Y}{K} \quad (3.21)$$

7 - Further more η_1 can be found by substituting (3.18) for λ_1, λ_2 and (3.19) for η_2 in the above equation

$$\eta_1=\frac{u_{wall2}-\frac{B\sigma_Y}{K}(1-2\frac{\sigma_Y L}{B(p_0-p_L)})+\frac{(p_0-p_L)B^2}{KL}(0.5-\frac{\sigma_Y L}{B(p_0-p_L)})}{\frac{(p_0-p_L)B^2}{KL}-2\frac{B\sigma_Y}{K}} \quad (3.22)$$

Once η_1 is determined η_2 can be calculated from (3.19), λ_1, λ_2 can be calculated from (3.18), u_{max} from (3.17) and C_3^C from (3.16). When these constants have been determined, a function for the velocity distribution can be written as follows

$$u(\eta)=\begin{cases}\frac{(p_0-p_L)B^2}{KL}(\lambda_1\eta-0.5\eta^2)-\frac{B\sigma_Y}{K}\eta, & 0\le\eta<\eta_1\\ u_{max}, & \eta_1\le\eta<\eta_2\\ \frac{(p_0-p_L)B^2}{KL}(\lambda_2\eta-0.5\eta^2)+\frac{B\sigma_Y}{K}\eta+C_3^C, & \eta_2\le\eta\le 1\end{cases} \quad (3.23)$$

3.6.3 Analytical solution for a Power Law Fluid

To obtain an analytical solution for the power law fluid the power law fluid model (3.1) needs to be inserted into equation (3.4). Once this done the following is obtained:

$$m\left|\frac{du}{dy}\right|^{n-1}\frac{du}{dy}=\frac{(p_0-p_L)B}{L}(\lambda-\eta) \quad (3.24)$$

After a similar procedure as for the Bingham fluid model a solution for the velocity profile can be found. Details to the solution to this problem can be found in Flumerfelt *et al.* (1969). The solution has the following procedure:

For a given value of the channel width B, length of the channel L, power law index n, fluid consistency index m, pressure at the beginning of the channel p_0, pressure at the end p_L and velocity at the moving wall u_{wall2} calculate:

$$\Lambda = \frac{(p_0 - p_L)B}{mL}\left(\frac{B}{u_{wall2}}\right)^n$$
$$s = 1/n$$

Using Λ and s a value for λ needs to be obtained. Depending on the value of Λ there are two possible equations for determining λ:

CASE I, $|\Lambda| \leq (s+1)^{(1/s)}$

$$f_I(\lambda) = \left(\frac{s+1}{\lambda^{s+1} - (\lambda - 1)^{s+1}}\right)^{1/s} - \Lambda = 0$$

CASE II, $|\Lambda| > (s+1)^{(1/s)}$

$$f_{II}(\lambda) = \left(\frac{s+1}{\lambda^{s+1} - (1 - \lambda)^{s+1}}\right)^{1/s} - \Lambda = 0$$

To determine the value for λ from the above equations a Newton-Raphson procedure needs to be employed. In CASE I the initial value needs to be $\lambda_{initial} = 50.0$ and in CASE II the initial value needs to be $\lambda_{initial} = 50.0$. The derivatives that are needed for the Newton-Raphson procedure have the following form:

$$f_I'(\lambda) = -(s+1)^{1/s}\frac{[(s+1)\left(\lambda^s - (\lambda - 1)^s\right)]}{[\lambda^{s+1} - (\lambda - 1)^{s+1}]^{\frac{s+1}{s}}}$$
$$f_{II}'(\lambda) = -(s+1)^{1/s}\frac{[(s+1)\left(\lambda^s + (1 - \lambda)^s\right)]}{[\lambda^{s+1} - (1 - \lambda)^{s+1}]^{\frac{s+1}{s}}}$$

Once λ is determined velocity u can be calculated by using the following formula:

$$\boxed{u(\eta) = u_{wall2}\frac{\lambda^{s+1} - |\lambda - \eta|^{s+1}}{\lambda^{s+1} - |\lambda - 1|^{s+1}} \text{ where } \eta \in [0, 1]} \tag{3.25}$$

3.6.4 Remarks on analytical solutions

In sections 3.6.1, 3.6.2 and 3.6.3 the analytical solutions for a Couette flow have been presented. These solutions have been presented in a way that makes them easy to implement in a program, and convenient for comparison to numerical values.

This will be done in chapter 6 as part of the verification of the numerical implementation. As it has been illustrated in the previous sections solving non-Newtonian fluid flow is a complex, lengthy process even for simple flows. For more complicated fluid flow analytical solutions are either very complicated or impossible to obtain. This emphasises the need for obtaining robust numerical methods to solve complex real world problems. A robust numerical strategy based on the finite element method, that can be used to solve real world problems will be elaborated in following chapters.

Chapter 4

Finite Element Methods

The previous chapters describe the basic equations that govern non-Newtonian fluid flow. This chapter will elaborate on the ways of solving these equations. Solving these equations analytically is not a practical option as has been illustrated in Section 3.6. Thus the only option is to solve them numerically. This chapter describes solution of the governing equations for non-Newtonian fluid flow via the finite element method. The overview of the chapter is as follows: First an overview of the finite element method in context of fluid flow will be presented. Next the Petrov-Galerkin method will be introduced. Finally details of the finite element formulation will be elaborated. This formulation will then be used in the following chapter to illustrate a solution procedure for obtaining numerical solutions.

4.1 Introduction to modelling fluid flow using the finite element method

Fluid flow is governed by an initial boundary value problems (IBVP), which means that the problem is described in both spatial and time domain. Traditionally the finite element method is used to discretize the spatial domain and this approach is taken in this work. The time dependent part of the IBVP is modelled by discrete integration in time, making this approach a semi-discrete method. We note that techniques which use finite element discretization in both space and time domain have gained popularity in recent years. This approach has not been pursued here, however. The IBVP that governs fluid flow was derived in chapter 2 and the material models for various non-Newtonian fluids were described in chapter 3.

The governing equations describing flows of non-Newtonian fluids can be summarised as follows:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \nabla\cdot\boldsymbol{\sigma}(\mathbf{u},p) = \mathbf{f}, \qquad \text{in } \Omega, \forall t \in]0,T[, \tag{4.1}$$

$$\nabla\cdot\mathbf{u} = 0, \qquad \text{in } \Omega, \forall t \in]0,T[, \tag{4.2}$$

$$\mathbf{u} - \mathbf{g} = \mathbf{0}, \qquad \text{on } \Gamma_g, \forall t \in]0,T[, \tag{4.3}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n} - \mathbf{h} = \mathbf{0}, \qquad \text{on } \Gamma_h, \forall t \in]0,T[, \tag{4.4}$$

$$\mathbf{u}(t_0) = \mathbf{u}_0 \quad p(t_0) = p_0, \qquad \text{in } \Omega, \tag{4.5}$$

where the stress tensor $\boldsymbol{\sigma}$ is defined in the following way:

$$\boldsymbol{\sigma}(\mathbf{u},p) = -p\mathbf{I} + 2\mu(\dot{\gamma})\boldsymbol{\varepsilon}(\mathbf{u}), \tag{4.6}$$

$$\boldsymbol{\varepsilon} = \nabla^S(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right),$$

$$\mu = \mu(\dot{\gamma}) \text{ is given by the material model,}$$

$$\dot{\gamma} = (2\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon})^{1/2},$$

$$\mathbf{f} - \text{body force}$$

The domain Ω is such that $\Omega \subseteq \mathbb{R}^{n_{\text{dim}}}$ where n_{dim} denotes the number of spatial dimensions. The boundary of the domain Ω is Γ, such that $\Gamma_g \cup \Gamma_h = \Gamma$ and $\Gamma_g \cap \Gamma_h = \emptyset$.

The governing equations (4.1)–(4.6) can be classified as second order mixed hyperbolic-parabolic type partial differential equations. The equations behaves in a more parabolic way if the diffusion term $\nabla\cdot\boldsymbol{\sigma}(\mathbf{u},p)$ is dominant compared to the advection term $\mathbf{u}\cdot\nabla\mathbf{u}$. If the opposite is true then the equation behaves in a more hyperbolic way.

As a starting point for the finite element discretization the problem (4.1)–(4.6) is written in the variational (weak) form as follows:

$$\int_\Omega \mathbf{w}\cdot\left[\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) - \nabla\cdot\boldsymbol{\sigma}(\mathbf{u},p) - \mathbf{f}\right] d\Omega + \int_\Omega q\cdot[\nabla\cdot\mathbf{u}]\, d\Omega = 0 \tag{4.7}$$

where $\mathbf{w}$ and q are weighting or test functions. Equation (4.7) is then transformed by applying the divergence theorem to the diffusion term $\mathbf{w}\cdot[\nabla\cdot\boldsymbol{\sigma}]$ as follows

$$\int_\Omega \mathbf{w}\cdot[\nabla\cdot\boldsymbol{\sigma}]\, d\Omega = -\int_\Omega \nabla\mathbf{w} : \boldsymbol{\sigma}\, d\Omega + \int_\Gamma \mathbf{w}\cdot\boldsymbol{\sigma}\mathbf{n}\, d\Gamma$$

Since the stress tensor $\boldsymbol{\sigma}$ is symmetric $\nabla\mathbf{w} : \boldsymbol{\sigma} \equiv \nabla^S\mathbf{w} : \boldsymbol{\sigma} = \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}$. Since $w = 0$ on Γ_g and $\boldsymbol{\sigma}\mathbf{n} = \mathbf{h}$ on Γ_h it follows

$$\int_\Omega \mathbf{w}\cdot[\nabla\cdot\boldsymbol{\sigma}]\, d\Omega = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}\, d\Omega - \int_{\Gamma_h} \mathbf{w}\cdot\mathbf{h}\, d\Gamma \tag{4.8}$$

After substituting (4.8) into (4.7), the weak form of the problem (4.1)–(4.6) reads :

Find $\mathbf{u} \in (V)^{n_{sd}}$ and $p \in S$ such that

$$\int_\Omega \mathbf{w} \cdot \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - [\boldsymbol{\varepsilon}(\mathbf{w}) \, : \, \boldsymbol{\sigma}(\mathbf{u}, p)] \, d\Omega +$$
$$\int_\Omega q \cdot [\nabla \cdot \mathbf{u}] \, d\Omega - \int_{\Gamma_h} \mathbf{w} \cdot \mathbf{h} \, d\Gamma = \int_\Omega \mathbf{w} \cdot \mathbf{f} \, \mathrm{d}\Omega \quad \forall \mathbf{w} \in (W)^{n_{sd}}, \forall q \in Q \quad (4.9)$$

where $(V)^{n_{sd}}, S$ and $(W)^{n_{sd}}, Q$ are usually referred to as spaces of trial and test functions, respectively, and n_{sd} represents the number of spatial dimensions.

One of the advantages of transforming the original boundary value problem into a weak variational form is to increase the space in which the solution is sought. This enables solutions to be found for problems that do not have classical solutions. since the weak form requires less regularity conditions on the solution. Another advantage is that boundary conditions can be taken into account in a more natural way. For example, the boundary condition (4.4) becomes part of the weak form.

It is also important to note that when the original IBVP has classical solutions, these are also solutions of the weak form. The opposite holds if the solution of the weak form belongs to the space of functions in which the IBVP is defined.

4.1.1 The finite element method

The starting point for the finite element method (FEM) is the weak form of an IBVP given in (4.9). The finite method generates an approximate solution by using a general and systematic technique, which consists of using finite-dimensional subspaces for the trial and test spaces. The subspaces are obtained by partitioning the problem domain, creating a mesh of non-overlapping elements. Typically these elements are chosen as triangles or quadrilaterals in 2D, or tetrahedral and hexahedral elements in 3D. An example of a 2D mesh composed of triangular elements is shown in Figure 4.1.

The mesh consists of triangles $K_j, \quad j = 1, ..., n_{elements}$ and nodes $B, \quad B = 1, ..., n_{nodes}$. Thus the trial space $(V)^{n_{sd}}$ is approximated with $(V^h)^{n_{sd}}$ and the test space $(W)^{n_{sd}}$ is approximated with $(W^h)^{n_{sd}}$. The subspace V^h has the dimension M and an associated basis $\{N_1, N_2, ..., N_M\}$. This means that any function $\mathbf{u} \in V^h$ or $p \in S^h$ can be represented in a unique way as

$$\mathbf{u}(\mathbf{x}) = \sum_{B=1}^{M} \mathbf{U}_B N_B^{\mathbf{u}}(\mathbf{x}), \qquad p(\mathbf{x}) = \sum_{B=1}^{M} P_B N_B^{p}(\mathbf{x}), \qquad (4.10)$$

where $\mathbf{U}_B$ and P_B are coefficients. The basis functions N_B also known as the shape functions, are created in such a way that:

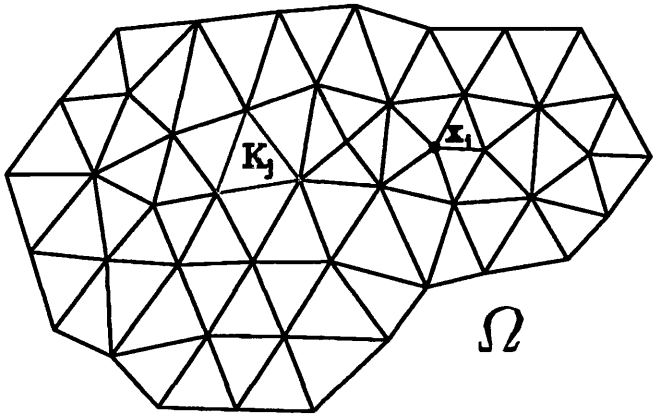


Figure 4.1: A 2D triangular mesh

$$N_B(\mathbf{x}_A) = \delta_{AB} = \begin{cases} 1 & A = B \\ 0 & A \neq B \end{cases}$$

where δ_{AB} is the Kronecker delta function. A graphical representation of the basis function can be found in Figure 4.2.

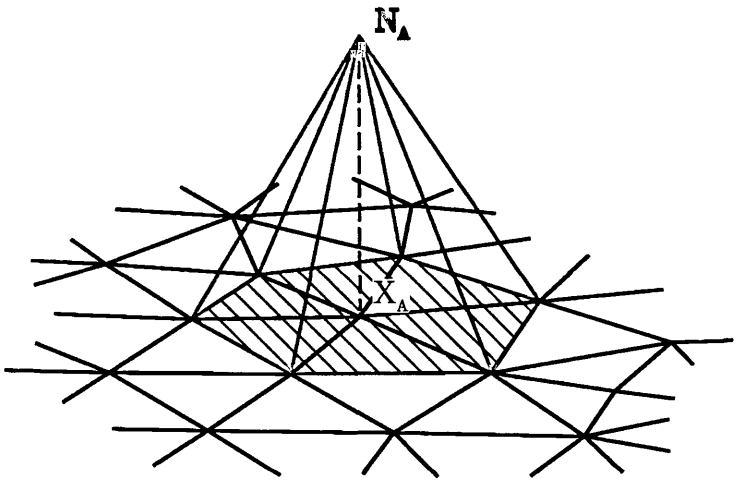


Figure 4.2: Shape function

Because of the way in which the basis functions are defined the values of the coefficients $\mathbf{U}_B$ and P_B are equal to the values of the functions at $\mathbf{x}_B$, i.e.

$$\mathbf{U}_B = \mathbf{u}(\mathbf{x}_B) \text{ and } P_B = p(\mathbf{x}_B).$$

Throughout this work a triangular element was chosen for 2D simulations and a tetrahedral element was chosen for 3D The chosen shape functions for velocity and pressure are linear. The elements that are used are shown in Figure 4.3.

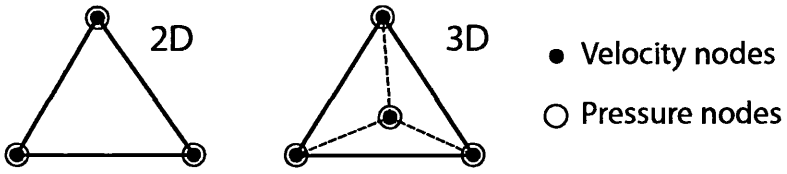


Figure 4.3: Velocity-pressure elements used

The chosen elements (Figure 4.3) are convenient for implementation into a standard finite element code because the unknown variables are associated with nodes. This is not true for some other elements used in this context such as the Raviart – Thomas finite elements (Girault & Raviart, 1979), for which the unknowns are not associated with the nodes.
The problem (4.9) can now be posed in the following way:

Find $\mathbf{u}^h \in (V^h)^{n_{sd}}$ and $p^h \in S^h$ such that

$$\int_\Omega \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \left[\boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \boldsymbol{\sigma}(u^h, p^h) \right] \, d\Omega +$$
$$\int_\Omega q^h \cdot \left[\nabla \cdot \mathbf{u}^h \right] \, d\Omega - \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h} \, d\Gamma = \int_\Omega \mathbf{w} \cdot \mathbf{f} \, \mathrm{d}\Omega \quad \forall \mathbf{w}^h \in (W^h)^{n_{sd}}, \forall q^h \in Q^h \tag{4.11}$$

where subscript h indicates that the relevant variables belong to a finite dimensional space. By substituting the trial functions with their finite basis representation the problem of determining the unknown functions $\mathbf{u}$ and P is transformed into one of determining values for the constants $\mathbf{U}_B$ and P_B. The weak form (4.11) for a given value of test functions yields an ordinary differential equation with unknown constants $\mathbf{U}_B$ and $\dot{\mathbf{U}}_B$, $B = 1, ..., m_{\mathbf{U}}$ and P_B, $B = 1, ..., m_P$. To be able to uniquely determine the unknown constants $m = m_{\mathbf{U}} + m_P$ different equations are required. This is done by choosing m linearly independent values for the test functions. After choosing test functions $\mathbf{w}_A^h$ and q_A^h the weak form is represented in the following way:

$$\int_\Omega \mathbf{w}_A^h \cdot \rho \left(\sum_{B=1}^{m_{\mathbf{U}}} \dot{\mathbf{U}}_B N_B^{\mathbf{u}} + \sum_{B=1}^{m_{\mathbf{U}}} \mathbf{U}_B N_B^{\mathbf{u}} \cdot \sum_{B=1}^{m_{\mathbf{U}}} \mathbf{U}_B \otimes \nabla N_B^{\mathbf{u}} \right) -$$
$$\left[\boldsymbol{\varepsilon}(\mathbf{w}_A^h) \, : \, \boldsymbol{\sigma} \left(\sum_{B=1}^{m_{\mathbf{U}}} \mathbf{U}_B N_B^{\mathbf{u}}, \sum_{i=B}^{m_P} P_B N_B^p \right) \right] \, d\Omega +$$
$$\int_\Omega q_A^h \cdot \left[\sum_{B=1}^{m_{\mathbf{U}}} \mathbf{U}_B \cdot \nabla N_B^{\mathbf{u}} \right] \, d\Omega - \int_{\Gamma_h} \mathbf{w}_A^h \cdot \mathbf{h} \, d\Gamma = \int_\Omega \mathbf{w}_A^h \cdot \mathbf{f} \, \mathrm{d}\Omega \tag{4.12}$$

4.1.2 Time integration

The time dependent part of the weak form (4.12) is discretized by using a time stepping scheme. The time stepping scheme that has been adopted belongs to the family of "one time step methods", which includes also the backward Euler as a special case (Simo & Armero, 1994). As a starting point for the time discretization scheme we introduce the following notation:

$$\begin{aligned} \mathbf{U}^{n+\alpha} &= \alpha \mathbf{U}^{n+1} + (1-\alpha)\mathbf{U}^{n} \\ P^{n+\alpha} &= \alpha P^{n+1} + (1-\alpha)P^{n} \end{aligned} \qquad \text{where } \alpha \in [0,1] \tag{4.13}$$

where the superscripts n, $n+1$ and $n+\alpha$ represent the values of the unknown variables at corresponding time instants t_n, t_{n+1} and $t_{n+\alpha}$, respectively. In the proposed discretization scheme the velocity gradient in the convection term is approximated by using a new parameter β, i.e.

$$\mathbf{U}^{n+\beta} = \beta \mathbf{U}^{n+1} + (1-\beta)\mathbf{U}^{n} \qquad \text{where } \beta \in [0,1]$$

By introducing the time discretization the weak form (4.12) results in

$$\begin{aligned} &\int_\Omega \mathbf{w}_A^h \cdot \rho \left(\sum_{B=1}^{m_\mathbf{U}} \frac{\mathbf{U}_B^{n+1} - \mathbf{U}_B^{n}}{\Delta t} N_B^{\mathbf{u}} + \sum_{B=1}^{m_\mathbf{U}} \mathbf{U}_B^{n+\alpha} N_B^{\mathbf{u}} \cdot \sum_{B=1}^{m_\mathbf{U}} \mathbf{U}_B^{n+\beta} \otimes \nabla N_B^{\mathbf{u}} \right) - \\ &\qquad \left[\boldsymbol{\varepsilon}(\mathbf{w}_A^h) \,:\, \boldsymbol{\sigma} \left(\sum_{B=1}^{m_\mathbf{U}} \mathbf{U}_B^{n+\alpha} N_B^{\mathbf{u}}, \sum_{i=B}^{m_P} P_B^{n+\alpha} N_B^{p} \right) \right] d\Omega + \\ &\int_\Omega q_A^h \cdot \left[\sum_{B=1}^{m_\mathbf{U}} \mathbf{U}_B^{n+\alpha} \cdot \nabla N_B^{\mathbf{u}} \right] d\Omega - \int_{\Gamma_h} \mathbf{w}_A^h \cdot \mathbf{h}^{n+\alpha} \, d\Gamma = \int_\Omega \mathbf{w}_A^h \cdot \mathbf{f}^{n+\alpha} \, \mathrm{d}\Omega \end{aligned}$$

Table 4.1 gives a summary of various time stepping schemes that are obtained for different values of α and β.

α	β	Name	Order of accuracy
1.0	1.0	Backward Euler	first
0.5	0.5	Crank-Nicholson	second
1.0	0.0		first

Table 4.1: Typical values for α and β

Since $\mathbf{w}_A^h$ and q_A^h are arbitrary the standard argument (Hughes *et al.*, 1987) gives a non-linear incremental algebraic problem which can be expressed in the following way:

$$\mathbf{K}_A(, ..., \mathbf{U}_B^{n+1}, ..., P_B^{n+1}, ...) = \mathbf{F}_A \qquad \text{where } A = 1, ..., m$$

For many problems involving non-Newtonian fluid flow the time dependent part can be neglected and a steady - state solution is sought. A flow chart that represents the various stages in the FEM is represented in Figure 4.4.

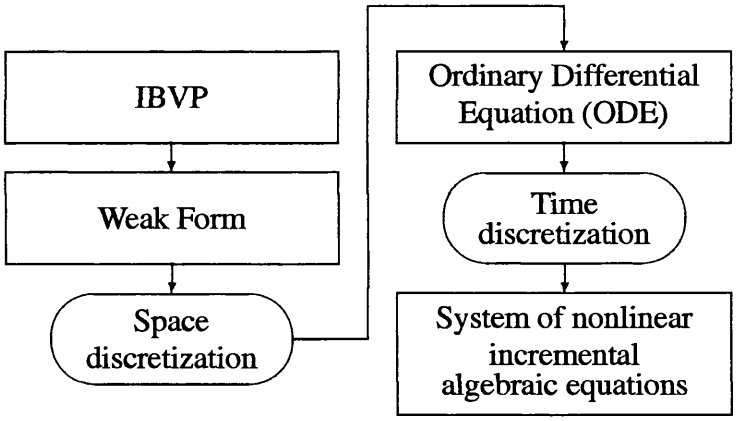


Figure 4.4: FEM

We note that the test functions can be chosen in a variety of ways. The most obvious choice is to make them identical to the trial functions. In this case the Galerkin finite element method (GFEM) is obtained. The Galerkin method has proved to be very effective for elliptic problems, such as those arising in elasticity. The equations that govern non-Newtonian fluid flow are of a hyperbolic-parabolic type, which makes the Galerkin method inappropriate (see (Johnson, 1995)). One of the ways to overcome the problems that occur with application of the Galerkin method for fluid flow problems is to use additional terms in the test functions. In this way the so-called Petrov-Galerkin finite element approach is obtained which is described in the next section.

4.2 The Petrov-Galerkin method

The Petrov-Galerkin method uses test functions that are not equal to the trail functions as a device for enhancing stability of the standard Galerkin method without upsetting consistency (Franca, 2001; Hughes *et al.*, 1986a). The original finite element formulation has two sources of instability. The first is related to occurrence of spurious node-to-node oscillations and stems from the hyperbolic-parabolic nature of the fluid equations. The second is related to the problem of pressure stabilisation and is related to the choice of interpolation functions for velocity and pressure. Both of these issues will be addressed in the Petrov-Galerkin manner by adding additional terms to the standard Galerkin weighting functions.

4.2.1 Streamline upwind stabilisation

It has been observed (Brooks & Hughes, 1982) that when the equations are solved by using a standard Galerkin method spurious node-to-node oscillations occur that have no physical meaning. The source of these oscillations can be identified by a reverse procedure. The procedure consists of taking a Galerkin discretization and transforming it back into a partial differential equation (Donea, 1983). If this partial differential equation is compared to the original partial differential equation, the conclusion that can be drawn is that the Galerkin method leads to a system that has less diffusion, thus this difference in the diffusion can be identified as the source of node-to-node oscillations. These oscillations are often referred to as wiggles. An example of how wiggles can render the solution useless can be seen in Figure 4.5.

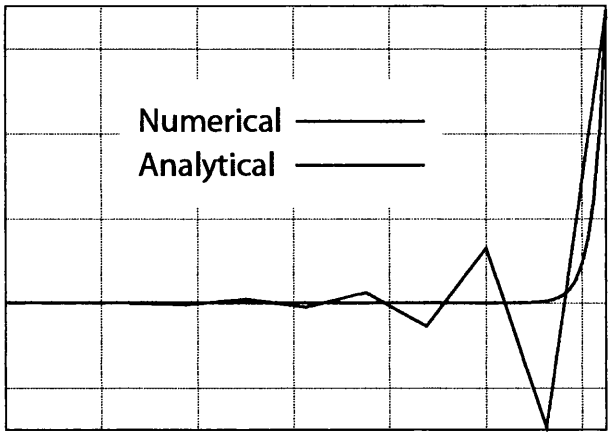


Figure 4.5: Numerical and analytical solution for an one dimensional advection–diffusion problem (Brooks & Hughes, 1982)

Wiggles are much more apparent with problems that are dominated by convection. When analysing convection diffusion type problems it is usual to define a coefficient between convection terms on one hand and diffusion terms on the other. This coefficient for fluids is called the Reynolds number *Re*. A high Reynolds number indicates a convection dominated problem. Unlike with Newtonian fluids, it is not in general possible to define a unique Reynolds number for the whole problem when considering non-Newtonian fluids. Wiggles occur most often when a boundary condition forces a rapid change of the solution, i.e. the solution is not smooth. One way of solving the problem of wiggles is to refine the mesh in areas where they occur (Gresho & Sani, 1998). This is not always desirable since the amount of elements should be kept to a minimum. The other way is to find ways of suppressing the wiggles, i.e. stabilising the solution.

Wiggles also appear in finite difference schemes since they approximate differential operators in the similar way. This means that various parallels can be drawn. The traditional way of solving this kind of problem in finite difference schemes is to add artificial diffusion. The reason for doing this is that diffusion

suppresses wiggles. The downside may be a loss of accuracy and the occurrence of the so called cross wind diffusion (Brooks & Hughes, 1982).

In order to remove the wiggles the so-called upwind difference scheme can be employed (Zienkiewicz & Taylor, 2000; Hughes, 1979). The problem with upwind differencing is that it is only first order accurate and it sometimes does not give a satisfactory solution. In terms of diffusion the upwind scheme is over diffusive whereas the central difference scheme can be categorised as an under diffusive scheme. With respect to the amount of diffusion the optimal scheme would seem to be one that lies between the central difference scheme and the purely upwind scheme. This amounts to adding the correct amount of diffusion in the direction of flow (Brooks & Hughes, 1982). If too much diffusion is added the wiggles are suppressed but the solution that is obtained may be wrong and does not truly represent the problem. If too little diffusion is added wiggles occur.

The amount of artificial diffusion that needs to be added can be determined in various ways. Either via a Galerkin least square approximation or via a balancing method (Zienkiewicz & Taylor, 2000; Hughes, 1979). The amount of artificial diffusion can also be determined by analysing the numerical error at nodal points. Once the amount is determined, diffusion can be applied in a variety of ways among which are modifying the quadrature rule to take into account the extra diffusion or modifying the weighting function. The second method has been chosen in this thesis thus making this a Petrov-Galerkin method (Brooks & Hughes, 1982).

The modification of the weighting function is performed by using the standard weighting function and adding to it a term that weights the upwind element more strongly than the downwind element. The weighting function has the following form:

$$\tilde{w} = \underbrace{w}_{\substack{\text{Galerkin} \\ \text{weighting}}} + \underbrace{\delta}_{\substack{\text{Upwind} \\ \text{weighting}}}$$

where the upwind weighting δ is constructed in such away that it emphasises upwind elements.

4.2.2 Pressure stabilisation

Pressure stabilisation is necessary to suppress large oscillations that occur in the solutions for the pressure field. These oscillations can be identified by the "checkerboard" effect they create, and are a result of the ill-posedness of the discrete problem obtained from the use of equal order interpolation function for velocity and pressure in discretization of the mixed formulation (4.12).

The term *mixed methods* is used in general to describe finite element methods that involve the approximation of two or more vector or scalar fields that are different in nature and defined on the same physical domain. In this case considerations of approximability of the exact solution by the finite element subspaces and

especially of the stability of the resulting discretization, in general, call for special observations on the choice of the interpolating spaces.

With this regard in mind an overview of the basic results on the existence, uniqueness and stability of the solution of some classical variational formulations will be given. This will also help to clarify the motivation behind the stabilised methods (Brezzi & Fortin, 1991).

Let V be a Hilbert space, $a(u,v)$ a bilinear form defined on $V \times V$ and f a linear continuous functional on V. We can consider the following variational problem:

$$\begin{aligned} &\text{Find } u \in V \\ &\quad a(u,v) = f(v) \qquad \forall v \in V \end{aligned} \tag{4.14}$$

A sufficient condition for this problem to have solution and to be unique is given by the Lax–Milgram theorem. This theorem states that problem (4.14) has solution and is unique if the bilinear form is ***continuous*** and ***coercive (or V-elliptic).*** The coercive property of the bilinear form reads as follows:

$$\exists \alpha > 0 : \quad a(v,v) \geq \alpha \|v\|^2 \qquad \forall v \in V \tag{4.15}$$

Also, theorem gives the following stability result:

$$\|u\| \leq \frac{1}{\alpha} \|f\| \tag{4.16}$$

which follows from:

$$\|f\| \stackrel{def}{=} \sup_{v \in V} \frac{|f(v)|}{\|v\|} \geq \frac{|f(u)|}{\|u\|} = \frac{|a(u,u)|}{\|u\|} \geq \alpha \|u\|$$

The conforming finite element approximation of problem (4.14) can be realised by using *any* finite dimensional subspace V_h of V. Then the approximation problem will read as follows:

$$\begin{aligned} &\text{Find } u_h \in V_h : \\ &\quad a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h \end{aligned} \tag{4.17}$$

If $a(u,v)$ is continuous and coercive on $V \times V$, these properties carry over on $V_h \times V_h$, that is, the bilinear form $a(u,v)$ is continuous and coercive also on $V_h \times V_h$ with the same constant α. As a result, the Lax-Milgram theorem applies and the independence of α on h plays an important role in the proof of the convergence of the finite element approximation (Ciarlet, 1978).

In studying a generalisation of problem (4.14) with a bilinear form defined over the Cartesian product of two in general different Hilbert spaces, Babuška in

(Babuška, 1973) replaces the coerciveness condition with the so-called inf–sup condition. The variational problem reads now as follows:

$$\begin{aligned}&\text{Given } a(u,v) \text{ defined on } U \times V\\ &\quad f(v) \text{ linear continuous functional on } V\\ &\text{Find } u \in U:\\ &\quad a(u,v) = f(v) \qquad \forall v \in V. \end{aligned} \tag{4.18}$$

A sufficient condition that guarantees existence and uniqueness of the solution is given by the continuity of the bilinear form $a(u,v)$ and the inf–sup condition, that is,

$$\exists \alpha > 0: \quad \sup_{v \in V} \frac{a(u,v)}{\|v\|} \geq \alpha \|u\| \qquad \forall u \in U \tag{4.19}$$

which is important for the stability of the solution.

Example Given the bilinear form $a(u,v)$ defined over $V \times V$, if $a(u,v)$ is coercive, then the bilinear form satisfies inf–sup condition. This is obtained as follows:

$$\forall u \in V$$

$$\sup_{v \in V} \frac{a(u,v)}{\|v\|} \overset{(a)}{\geq} \frac{a(u,v)}{\|v\|} \geq \frac{a(u,u)}{\|u\|} \overset{(b)}{\geq} \frac{\alpha \|u\|^2}{\|u\|} = \alpha \|u\|$$

The implication (a) follows from the definition of supremum whereas the implication (b) from the coerciveness of the bilinear form ∎

A finite element approximation of problem (4.18) reads as follows

$$\begin{aligned}&\text{Given } U_h \subset U \text{ and } V_h \subset V\\ &\text{Find } u_h \in U_h:\\ &\quad a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h \end{aligned} \tag{4.20}$$

Unlike problem (4.17) where the coercivity of the bilinear form was preserved by replacing V_h with V, the inf–sup condition in general does not carry over by considering finite dimensional subspaces of the original spaces, so that the discrete scheme (4.20) can result in an ill–posed problem. In order to guarantee the solvability of the approximate scheme (4.20), the spaces U_h and V_h must therefore be chosen so that the given bilinear form $a(u,v)$ satisfies condition (4.19). As a by product of the meeting of this condition, we also obtain an optimal error estimate which proves the convergence of the approximate scheme.

The same inf–sup condition has been obtained by Brezzi in (Brezzi, 1974) by analysing the minimisation of a quadratic functional with linear constraints which

delivers the following:

Given $a(u,w)$ defined on $W \times W$
$b(w,q)$ defined on $W \times Q$
$f(w)$ linear continuous functional on W
$g(q)$ linear continuous functional on Q
Find $(u,p) \in W \times Q$:

$$\begin{aligned} a(u,w) + b(w,p) &= f(w) \quad \forall w \in W. \\ b(u,q) &= g(q) \quad \forall q \in Q. \end{aligned} \tag{4.21}$$

A sufficient condition that guarantees the existence and uniqueness of the solution is represented by the continuity of the bilinear forms $a(u,w)$ and $b(w,q)$; the coerciveness of $a(u,v)$ on a suitable subspace of $W \times W$ and the following stability condition

$$\exists \alpha > 0 : \sup_{w \in W} \frac{b(w,q)}{\|w\|_W} \geq \alpha \|q\|_Q \quad \forall q \in Q \tag{4.22}$$

For the finite element approximation of problem (4.21) the same considerations discussed in relation to the problem (4.18) apply, due to the presence of the inf–sup condition (4.22) (Brezzi & Fortin, 1991). The latter imposes very strict restrictions on the choice of the finite dimensional subspaces $W_h \subset W$ and $Q_h \subset Q$. There are, however, examples of finite element spaces which pass the inf–sup condition but they are difficult to implement (Girault & Raviart, 1979).

Another philosophy in proposing an approximation of the problem (4.18) and (4.22) has been therefore to choose finite element spaces convenient from an implementation standpoint and to modify the bilinear and linear forms so that first of all a certain consistency condition is met, and the resulting problem has solution, which is unique and stable. This is basically the underlining idea of the so-called stabilised method for mixed formulations. Following the work of (Tezduyar *et al.*, 1992), the choice of the weighting function as

$$\tilde{w} = \underset{\substack{\text{Galerkin} \\ \text{weighting}}}{w} + \underset{\substack{\text{Upwind} \\ \text{weighting}}}{\delta} + \underset{\substack{\text{Pressure} \\ \text{stabilising}}}{\epsilon}$$

modifies indeed the original formulation so that the consistency is preserved and the stability condition on the given interpolating spaces is easier to realise (Hughes *et al.*, 1986a).

In the next section the above weighting function will be applied to the variational formulation of the fluid flow. This formulation will then serve as a starting point for solving non-Newtonian fluid flow problems via the finite element method.

4.3 Formulation

As outlined in the previous sections the process of obtaining a solution via FEM consists of posing the problem in a weak form and then discretizing the spaces for the test and trial functions. The Sobolev space will be used for approximating the trial and test functions. The Sobolev space is a Hilbert space. To be able to define a Sobolev space the square integrable functions and a multi-index notation for differential operators need to be defined. It is said for a function u that it is a square integrable function on Ω if the following condition is satisfied $\int_\Omega u^2 d\Omega < \infty$. Let $L^2(\Omega)$ denote the space of all the square integrable functions on Ω. Then $L^2(\Omega)$ can be written as:

$$L^2(\Omega) = \left\{ u \,|u \in \Omega \,, \int_\Omega u^2 d\Omega < \infty \right\}$$

A multi-index notation for differential operators is defined in the following way:

$$D^\alpha u = \frac{\partial^{|\alpha|} u}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_n^{\alpha_n}}$$

where α is an ordered n-tuple of nonnegative integers $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $|\alpha|$ is the sum $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$. This means that if $|\alpha| = m$ then $D^\alpha u$ represents one of the mth partial derivative of u.

The Sobolev space $H^m(\Omega)$ of order m can be defined as a space of all functions that belong to the space $L^2(\Omega)$ together with all their weak partial derivatives up to and including m, i.e.

$$H^m(\Omega) = \left\{ u \,\middle|\, D^\alpha u \in L^2(\Omega) \,, \text{ for all } \alpha \text{ such that } |\alpha| \leq m \right\}$$

More details about Sobolev spaces can be found in Temam (1977); Johnson (1995); Reddy (1998).

The discretized trial functions spaces S_u^h and S_p^h and the discretized test functions spaces V_u^h, V_p^h that are used in the formulation can be defined as follows:

$$S_u^h = \left\{ \mathbf{u}^h \,\middle|\, \mathbf{u}^h \in (H^{1h})^{n_{sd}} \,, \mathbf{u}^h = \mathbf{g}^h \quad \text{on} \quad \Gamma_g \right\}, \tag{4.23}$$

$$V_u^h = \left\{ \mathbf{w}^h \,\middle|\, \mathbf{w}^h \in (H^{1h})^{n_{sd}} \,, \mathbf{w}^h = \mathbf{0} \quad \text{on} \quad \Gamma_g \right\}, \tag{4.24}$$

$$S_p^h = \left\{ p^h \,\middle|\, p^h \in H^{1h} \right\}, \tag{4.25}$$

$$V_p^h = \left\{ q^h \,\middle|\, q^h \in H^{1h} \right\}, \tag{4.26}$$

where $\mathbf{u}^h$ represents the discretized velocity, p^h the discretized pressure, $\mathbf{w}^h$ and q^h are the weighting functions, H^{1h} represents a finite dimensional Sobolev space of order 1, n_{sd} is the number of spatial dimensions, Γ_g as before represents the part of the boundary where the velocities are defined.

The weak form of the fluid flow problem described in Box 2.1 can now be stated as follows:

$$\int_{\Omega_e} \tilde{\mathbf{w}}^h \cdot \left[\rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \boldsymbol{\sigma}^h(\mathbf{u}^h, p^h) - \mathbf{f} \right] d\Omega + \int_{\Omega_e} q^h \cdot \left[\nabla \cdot \mathbf{u}^h \right] d\Omega = 0$$

where the Petrov-Galerkin weighting function is defined as:

$$\tilde{\mathbf{w}}^h = \mathbf{w}^h + \boldsymbol{\delta}^h + \boldsymbol{\epsilon}^h \tag{4.27}$$

The streamline upwind weighting function $\boldsymbol{\delta}^h$ and the pressure weighting function $\boldsymbol{\epsilon}^h$ are defined as

$$\boldsymbol{\delta}^h = \tau(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \tag{4.28}$$

$$\boldsymbol{\epsilon}^h = \tau \frac{1}{\rho} \nabla q^h \tag{4.29}$$

The stabilising parameter τ scales the stabilisation contribution to the element depending on the element Reynolds number Re_u and the velocity norm $\|\mathbf{u}^h\|$, and is defined as

$$\tau = \frac{h_{len}}{2\|\mathbf{u}^h\|} z(Re_u) \qquad z(Re_u), = \frac{1}{\sqrt{1 + \frac{9}{Re_u^2}}},$$

where h_{len} represents the characteristic element length. For 2D problems it is adopted to be $h_{len} = \sqrt{2A}$ where A is equal to area of the element. The characteristic element length in 3D is $h_{len} = \sqrt[3]{V}$.

The element Reynolds number is defined in terms of velocity norm $\|\mathbf{u}^h\|$, element length h_{len}, density ρ and the viscosity μ. It should be noted that unlike the case of Newtonian fluids the viscosity is not constant for non-Newtonian fluids and in general is different for every element,

$$Re_u = \frac{\|\mathbf{u}^h\| h_{len} \rho}{2\mu(\mathbf{u}^h)}.$$

By employing the definition of the weighting function (4.27) the following weak form is obtained:

$$\begin{aligned}
&\int_{\Omega_e} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega - \int_{\Omega_e} \mathbf{w}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h \right] d\Omega + \\
&\sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\delta}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\delta}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h \right] d\Omega + \\
&\sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\epsilon}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\epsilon}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h \right] d\Omega + \\
&\int_{\Omega_e} q^h \cdot \left[\nabla \cdot \mathbf{u}^h \right] d\Omega = \int_{\Omega} \tilde{\mathbf{w}}^h \cdot \mathbf{f} \, \mathrm{d}\Omega
\end{aligned} \tag{4.30}$$

Following the standard FE formulation Green's formula is applied to the terms involving stress divergence $\nabla \cdot \boldsymbol{\sigma}^h$.

If the Green's formula is applied to the second term $\mathbf{w}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h\right]$ the following is obtained:

$$\begin{aligned}\int_\Omega \mathbf{w}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h\right] \, d\Omega &= -\int_\Omega \nabla \mathbf{w}^h \, : \, \boldsymbol{\sigma}^h \, d\Omega + \int_\Gamma \mathbf{w}^h \cdot \boldsymbol{\sigma}^h \mathbf{n} \, d\Gamma \\ &= -\int_\Omega \nabla \mathbf{w}^h \, : \, \boldsymbol{\sigma}^h \, d\Omega + \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, d\Gamma\end{aligned}$$

By employing the fact that $\boldsymbol{\sigma}^h$ is symmetric and using the boundary conditions (4.3) it follows

$$\boxed{\int_\Omega \mathbf{w}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h\right] \, d\Omega = \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \boldsymbol{\sigma}^h \, d\Omega - \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, d\Gamma} \tag{4.31}$$

When the result of the application of Green's formula is substituted back into (4.30) we get

$$\begin{aligned}&\int_{\Omega_e} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) \, d\Omega + \int_{\Omega_e} \boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \boldsymbol{\sigma}^h \, d\Omega + \\ &\sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\delta}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) \, d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\delta}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h\right] \, d\Omega + \\ &\sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\epsilon}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) \, d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\epsilon}^h \cdot \left[\nabla \cdot \boldsymbol{\sigma}^h\right] \, d\Omega + \\ &\int_{\Omega_e} q^h \, \nabla \cdot \mathbf{u}^h \, d\Omega = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, d\Gamma + \int_\Omega \tilde{\mathbf{w}}^h \cdot \mathbf{f} \, \mathrm{d}\Omega\end{aligned} \tag{4.32}$$

The diffusion term can be further expanded using the constitutive equation (4.6). The stabilisation terms are expanded in the same way by using

$$\nabla \cdot \boldsymbol{\sigma}^h = \nabla \cdot (-p^h \mathbf{I} + 2\mu \boldsymbol{\varepsilon}^h) = -\nabla p^h + \nabla \cdot 2\mu \boldsymbol{\varepsilon}^h$$

When this is substituted back into the formulation the following expression is

obtained

$$\int_{\Omega_e} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega -$$
$$\int_{\Omega_e} p^h \, \boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \mathbf{I} \, d\Omega + \int_{\Omega_e} 2\mu \, \boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \boldsymbol{\varepsilon}(\mathbf{u}^h) \, d\Omega +$$
$$\sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho \boldsymbol{\delta}^h \cdot \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\delta}^h \cdot \nabla p^h \, d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\delta}^h \cdot \left(\nabla \cdot 2\mu \boldsymbol{\varepsilon}^h \right) +$$
$$\sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho \boldsymbol{\epsilon}^h \cdot \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\epsilon}^h \cdot \nabla p^h \, d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \boldsymbol{\epsilon}^h \cdot \left(\nabla \cdot 2\mu \boldsymbol{\varepsilon}^h \right) +$$
$$\int_{\Omega_e} q^h \, \nabla \cdot \mathbf{u}^h \, d\Omega = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, d\Gamma + \int_{\Omega} \tilde{\mathbf{w}}^h \cdot \mathbf{f} \, \mathrm{d}\Omega \tag{4.33}$$

By substituting (4.28) and (4.29) into (4.33) the final formulation is obtained:

$$\int_{\Omega_e} \rho \mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t} \, d\Omega + \int_{\Omega_e} \rho \mathbf{w}^h \cdot (\mathbf{u}^h \cdot \nabla \mathbf{u}^h) \, d\Omega -$$
$$\int_{\Omega_e} p^h \, \boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \mathbf{I} \, d\Omega + \int_{\Omega_e} 2\mu \, \boldsymbol{\varepsilon}(\mathbf{w}^h) \, : \, \boldsymbol{\varepsilon}(\mathbf{u}^h) \, d\Omega +$$
$$\sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho \tau (\mathbf{u}^h \cdot \nabla) \mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t} \, d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho \tau (\mathbf{u}^h \cdot \nabla) \mathbf{w}^h \cdot (\mathbf{u}^h \cdot \nabla \mathbf{u}^h) \, d\Omega +$$
$$\sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau (\mathbf{u}^h \cdot \nabla) \mathbf{w}^h \cdot \nabla p^h \, d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau (\mathbf{u}^h \cdot \nabla) \mathbf{w}^h \cdot \left(\nabla \cdot 2\mu \boldsymbol{\varepsilon}^h \right) \, d\Omega +$$
$$\sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau \nabla q^h \cdot \frac{\partial \mathbf{u}^h}{\partial t} \, d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau \nabla q^h \cdot (\mathbf{u}^h \cdot \nabla \mathbf{u}^h) \, d\Omega +$$
$$\sum_{e=1}^{n_{el}} \int_{\Omega_e} \frac{\tau}{\rho} \nabla q^h \cdot \nabla p^h \, d\Omega - \sum_{e=1}^{n_{el}} \int_{\Omega_e} \frac{\tau}{\rho} \nabla q^h \cdot \left(\nabla \cdot 2\mu \boldsymbol{\varepsilon}^h \right) \, d\Omega +$$
$$\int_{\Omega_e} q^h \, \nabla \cdot \mathbf{u}^h \, d\Omega = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, d\Gamma + \int_{\Omega} \tilde{\mathbf{w}}^h \cdot \mathbf{f} \, \mathrm{d}\Omega \tag{4.34}$$

Once discretization is performed the above equation can be represented in a matrix form

$$\begin{aligned} \mathbf{M}\mathbf{a} + \mathbf{M}_\delta \mathbf{a} + \mathbf{N}(\mathbf{u}) + \mathbf{N}_\delta(\mathbf{u}) + \mathbf{K}(\mathbf{u}) - \mathbf{K}_\delta(\mathbf{u}) - \mathbf{G}\mathbf{p} + \mathbf{G}_\delta(\mathbf{u})\mathbf{p} &= \mathbf{F} + \mathbf{F}_\delta \\ \mathbf{M}_\epsilon \mathbf{a} + \mathbf{N}_\epsilon(\mathbf{u}) - \mathbf{K}_\epsilon(\mathbf{u}) + \mathbf{G}^T \mathbf{u} + \mathbf{G}_\epsilon \mathbf{p} &= \mathbf{E} + \mathbf{E}_\epsilon \end{aligned} \tag{4.35}$$

Here $\mathbf{u}$ is the vector of unknown values of $\mathbf{u}^h$, vector $\mathbf{a}$ represents the time derivative of $\mathbf{u}$ and $\mathbf{p}$ is the vector of unknown values of p^h. The matrices $\mathbf{M}$, $\mathbf{N}$, $\mathbf{K}$ and $\mathbf{G}$ on the LHS represent the time-dependent, advective, viscous and pressure terms respectively and have the following form

$$\mathbf{Ma} = \int_{\Omega_e} \rho \mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t}\, d\Omega \tag{4.36}$$

$$\mathbf{M}_\delta \mathbf{a} = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho \tau(\mathbf{u}^h)(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t}\, d\Omega \tag{4.37}$$

$$\mathbf{M}_\epsilon \mathbf{a} = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau(\mathbf{u}^h)\nabla q^h \cdot \frac{\partial \mathbf{u}^h}{\partial t}\, d\Omega \tag{4.38}$$

$$\mathbf{N}(\mathbf{u}) = \int_{\Omega_e} \rho \mathbf{w}^h \cdot (\mathbf{u}^h \cdot \nabla \mathbf{u}^h)\, d\Omega \tag{4.39}$$

$$\mathbf{N}_\delta(\mathbf{u}) = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho \tau(\mathbf{u}^h)(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot (\mathbf{u}^h \cdot \nabla \mathbf{u}^h)\, d\Omega \tag{4.40}$$

$$\mathbf{N}_\epsilon(\mathbf{u}) = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau(\mathbf{u}^h)\nabla q^h \cdot (\mathbf{u}^h \cdot \nabla \mathbf{u}^h)\, d\Omega \tag{4.41}$$

$$\mathbf{K}(\mathbf{u}) = \int_{\Omega_e} 2\mu\, \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\mathbf{u}^h)\, d\Omega \tag{4.42}$$

$$\mathbf{K}_\delta(\mathbf{u}) = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau(\mathbf{u}^h)(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot [\nabla \cdot 2\mu\boldsymbol{\varepsilon}(u^h)]\, d\Omega \tag{4.43}$$

$$\mathbf{K}_\epsilon(\mathbf{u}) = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \frac{\tau(\mathbf{u}^h)}{\rho}\nabla q^h \cdot [\nabla \cdot 2\mu\boldsymbol{\varepsilon}(u^h)]\, d\Omega \tag{4.44}$$

$$\mathbf{G}\mathrm{p} = \int_{\Omega_e} p^h\, \boldsymbol{\varepsilon}(\mathbf{w}^h) : \mathbf{I}\, d\Omega = \int_{\Omega_e} p^h \,\mathrm{tr}[\boldsymbol{\varepsilon}(\mathbf{w}^h)]\, d\Omega \tag{4.45}$$

$$\mathbf{G}_\delta(\mathbf{u})\mathrm{p} = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau(\mathbf{u}^h)(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot \nabla p^h\, d\Omega \tag{4.46}$$

$$\mathbf{G}_\epsilon(\mathbf{u})\mathrm{p} = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \tau(\mathbf{u}^h)\frac{1}{\rho}\nabla q^h \cdot \nabla p^h\, d\Omega \tag{4.47}$$

$$\mathbf{G}^T\mathbf{u} = \int_{\Omega_e} q^h\, \nabla \cdot \mathbf{u}^h\, d\Omega \tag{4.48}$$

The terms on the RHS of (4.35) $\mathbf{F}$, $\mathbf{F}_\delta$ are due to Dirichlet and Neumann type boundary conditions and also involve $\int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h\, d\Gamma$. The terms $\mathbf{E}$ and $\mathbf{E}_\epsilon$ come from the Dirichlet type boundary condition.

$$\mathbf{F} = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, d\Gamma + \int_{\Omega} \mathbf{w}^h \cdot \mathbf{f} \, d\Omega \tag{4.49}$$

$$\mathbf{F}_\delta = \int_{\Omega} \tau(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot \mathbf{f} \, d\Omega \tag{4.50}$$

$$\mathbf{E} = 0 \tag{4.51}$$

$$\mathbf{E}_\epsilon = \int_{\Omega} \tau \frac{1}{\rho} \nabla q^h \cdot \mathbf{f} \, d\Omega \tag{4.52}$$

Chapter 5

Solution Procedure

The space discretization by means of the finite element method described in chapter 4 transforms the IBVP that governs non-Newtonian fluid flow into a system of ordinary differential equations. These equations are then numerically integrated via a time stepping scheme to finally yield a system of non-linear algebraic equations. We can distinguish two types of non linearity. The first is the non-linearity of the advective term. This type of non-linearity is also present for Newtonian fluids and is the main source of non-linearity when solving Navier-Stokes equations. The second source of non linearity are the fluid material models that are specific to non-Newtonian fluids.

5.1 Review of time stepping schemes

The time stepping scheme that was adopted in this thesis belongs to a class of a single step methods. A single step method means that the transition from a time step n to the next time step $n+1$ is performed in a single step. This is one of simplest time stepping schemes and with the appropriate values of the integration parameters (see chapter 4) the scheme is unconditionally stable. There are several alternatives which will be reviewed in the following paragraphs.

One way of solving the system of non linear algebraic equations is to employ the so-called Picard iterative procedure. The procedure consists of fixing the viscosity to a previously known value and obtaining the solution. Then the velocity field is updated and the viscosity is recalculated and compared with the previous value. This process is repeated until convergence is obtained. This method was first proposed by Tanner *et al.* (1975) and further extended by O'Donovan & Tanner (1984). The extension was obtained by using not only the value of the viscosity in the previous iteration but also the value before the last. This in effect enables a type of extrapolation to take place which helps the convergence (O'Donovan & Tanner, 1984). The method gives results for power law fluids but for Bingham fluids it seems unsatisfactory. In both situations many iteration are needed to obtain convergence.

There are other approaches that are based on the so-called operator splitting methods. One of these methods was used by Baloch *et al.* (1995) in the solution of expansion flows. The method has three steps and can be either explicit or semi-implicit. In the first step a non-divergence free velocity field is computed in two sub steps. The second step a Poisson's equation is solved for pressure difference. The final step is used to determine a divergence free velocity field. Once these steps are discretized using the finite element method the solution is obtained by sequential advancement. Every step involves a solution of a linear set of equations. In this way the non-linearity of the problem is overcome.

Both above approaches are not suitable for the solution of non-Newtonian fluid flows. One of the reasons is that the material models are complex and have a profound influence on the nature of the fluid flow. This means that even if the non linearity of the advective terms is overcome by some of the above methods the non linearity of viscosity remains. Since it is not possible to avoid this non linearity a one step time stepping scheme has been chosen. To solve the non linear system the Newton-Raphson method has been chosen. The use of a Newton-Raphson procedure allows us to solve directly a system of highly nonlinear algebraic equations in one step without any need for further splitting. This method also enjoys rapid convergence when the given solution is in the appropriate neighbourhood of the solution. First the details of the discretization will be described.

5.2 Discretization of internal forces

In the previous chapter the finite element formulation was described. Now details on how the problem (4.34) is solved will be shown. There are two types of simulation that are carried out. All the different parts that contribute to the finite element formulation will be described in turn. The derivation of force vectors will be given first for 2D and then expanded for 3D case.

5.2.1 Internal Force - Part Ma

This part represents the time dependent component of the problem and is only present in transient simulations,

$$\mathbf{Ma} = \int_{\Omega} \rho \, \mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t} \, \mathrm{d}\Omega, \tag{5.1}$$

where the time derivative of $\mathbf{u}^h$ is defined via a simple finite difference operator

$$\frac{\partial \mathbf{u}^h}{\partial t} = \frac{\mathbf{u}^h_{n+1} - \mathbf{u}^h_n}{\Delta t}$$

The time stepping scheme assumes that the equations are posed at t_{n+1}. The velocity at time instant $n+1$ ($\mathbf{u}^h_{n+1}$) can be expressed in terms of $\mathbf{u}^h_{n+\alpha}$ and $\mathbf{u}^h_n$. The

following relationship is used for this purpose

$$\mathbf{u}^h_{n+\alpha} = \alpha \mathbf{u}^h_{n+1} + (1-\alpha)\mathbf{u}^h_n \tag{5.2}$$

This means that $\mathbf{u}^h_{n+1}$ can be defined as follows

$$\mathbf{u}^h_{n+1} = \frac{1}{\alpha}\left[\mathbf{u}^h_{n+\alpha} - (1-\alpha)\mathbf{u}^h_n\right]$$

This in turn makes the time derivative equal to

$$\frac{\partial \mathbf{u}^h}{\partial t} = \frac{\mathbf{u}^h_{n+1} - \mathbf{u}^h_n}{\Delta t} = \frac{\mathbf{u}^h_{n+\alpha} - \mathbf{u}^h_n}{\alpha \Delta t} \tag{5.3}$$

For every element the test function $\mathbf{w}^h$ and the trial function $\mathbf{u}^h$ are defined in the following way:

$$\mathbf{w}^h = \sum_{\mathrm{A}=1}^{NNodes} \mathrm{N_A}\mathbf{w_A} \tag{5.4}$$

$$\mathbf{u}^h_{n+\alpha} = \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}^{n+\alpha}_{\mathrm{B}} \tag{5.5}$$

$$\mathbf{u}^h_{n+\beta} = \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}^{n+\beta}_{\mathrm{B}} \tag{5.6}$$

$$\mathbf{u}^h_{n+1} = \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}^{n+1}_{\mathrm{B}} \tag{5.7}$$

$$\mathbf{u}^h_{n} = \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}^{n}_{\mathrm{B}} \tag{5.8}$$

where $\mathrm{N_A}$ and $\mathrm{N_B}$ are the shape functions, $\mathbf{u_B}$ is the velocity vector at node B and $\mathbf{w_A}$ is also a vector and is equal to the value of the test function at node A. The value of $NNodes$ is equal to the number of nodes in an element.

If values for the test and trial function are substituted back into (5.1) the value of **Ma** can be expressed as

$$\begin{aligned}
\mathbf{Ma} &= \int_\Omega \rho\, \mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t}\, \mathrm{d}\Omega \\
&\rightarrow \rho \int_\Omega \mathrm{N_A}\mathbf{w_A} \cdot \sum_{\mathrm{B}=1}^{NNodes} \frac{\mathrm{N_B}\mathbf{u}^{n+1}_{\mathrm{B}} - \mathrm{N_B}\mathbf{u}^{n}_{\mathrm{B}}}{\Delta t}\, \mathrm{d}\Omega \\
&= \rho \int_\Omega \mathrm{N_A}\mathbf{w_A} \cdot \sum_{\mathrm{B}=1}^{NNodes} \frac{\mathrm{N_B}\mathbf{u}^{n+\alpha}_{\mathrm{B}} - \mathrm{N_B}\mathbf{u}^{n}_{\mathrm{B}}}{\alpha \Delta t}\, \mathrm{d}\Omega
\end{aligned}$$

$$\begin{aligned}\mathbf{Ma} &= \frac{\rho}{\alpha\Delta t}\,\mathbf{w}_{\mathrm{A}}\cdot\sum_{\mathrm{B}=1}^{NNodes}\int_{\Omega}\mathrm{N_A N_B}\,\mathrm{d}\Omega(\mathbf{u}_{\mathrm{B}}^{n+\alpha}-\mathbf{u}_{\mathrm{B}}^{n}) \\ &= \frac{\rho}{\alpha\Delta t}\,\mathbf{w}_{\mathrm{A}}\cdot\sum_{\mathrm{B}=1}^{NNodes}\mathbf{M}_{\mathrm{AB}}(\mathbf{u}_{\mathrm{B}}^{n+\alpha}-\mathbf{u}_{\mathrm{B}}^{n})\end{aligned} \tag{5.9}$$

where the symbol $\rightarrow$ indicates that the global expressions of the LHS has been substituted for the local expressions. The value for $\mathbf{M}_{\mathrm{AB}}$ can be evaluated explicitly when the shape functions are linear for a triangular or tetrahedral element by using the following formulae (see Zienkiewicz & Taylor, 2000),

$$\begin{aligned}\int_{\Omega} L_1^m L_2^n L_3^p\,\mathrm{d}\Omega &= 2Ad\frac{m!n!p!}{(m+n+p+2)!} \quad \text{2D case} \\ \int_{\Omega} L_1^m L_2^n L_3^p L_4^q\,\mathrm{d}\Omega &= 6V\frac{m!n!p!q!}{(m+n+p+q+3)!} \quad \text{3D case}\end{aligned} \tag{5.10}$$

where L_i are the shape functions, A is the area of the triangle and V is the volume of the tetrahedral. When this formula is applied to our case $\mathbf{M}_{\mathrm{AB}}$ is equal to

$$\mathbf{M}_{\mathrm{AB}} = \begin{cases} 2Ad\frac{2!}{(2+2)!} = \frac{1}{6}Ad, & \mathrm{A}=\mathrm{B} \\ 2Ad\frac{1!1!}{(1+1+2)!} = \frac{1}{12}Ad, & \mathrm{A}\neq\mathrm{B}\end{cases} \quad \text{2D case}$$

$$\mathbf{M}_{\mathrm{AB}} = \begin{cases} 6V\frac{2!}{(2+2)!} = \frac{1}{2}V, & \mathrm{A}=\mathrm{B} \\ 6V\frac{1!1!}{(1+1+2)!} = \frac{1}{4}V, & \mathrm{A}\neq\mathrm{B}\end{cases} \quad \text{3D case}$$

where d is equal to the thickness in the 2D case. The thickness takes different values depending on whether the problem is plane stress/strain case or axisymmetric case. The value for d has the following values for different cases

$$d = \begin{cases} 1, & \text{Plane stress/strain} \\ 2\pi y_c, & \text{Axisymmetric about the X axis} \\ 2\pi x_c, & \text{Axisymmetric about the Y axis}\end{cases} \tag{5.11}$$

where x_c and y_c are coordinates of the centre of the triangle in 2D. In the 3D case they represent the centre of the tetrahedra. The values for x_c and y_c are defined as follows

$$x_c = \frac{1}{NNodes}\sum_{i=1}^{NNodes} x_i \quad \text{and} \quad y_c = \frac{1}{NNodes}\sum_{i=1}^{NNodes} y_i$$

where $NNodes$ represents the number of nodes.

In an attempt to further simplify and optimise the process of transforming equations into code it is helpful to isolate the constant part of equation (5.9). The constant part of the expression $C_{\mathrm{AB}}^{\mathbf{Ma}}$ can be defined in the following table

C_{AB}^{Ma}	2D	3D
$A = B$	$\frac{\rho A d}{6\alpha\Delta t}$	$\frac{\rho V}{2\alpha\Delta t}$
$A \neq B$	$\frac{\rho A d}{12\alpha\Delta t}$	$\frac{\rho V}{4\alpha\Delta t}$

Table 5.1: Values for C_{AB}^{Ma}

This makes the value of the force vector at node A equal to

$$\mathbf{F}_{A}^{Ma} = \sum_{B=1}^{NNodes} C_{AB}^{Ma}(\mathbf{u}_{B}^{n+\alpha} - \mathbf{u}_{B}^{n}) \tag{5.12}$$

F will be used in the future to denote force vectors that corresponds to velocity variables.

5.2.2 Internal Force - Part $\mathbf{N(u)}$

This is the part that represents the convective part of the problem and is present in both transient and steady state solutions,

$$\mathbf{N}(\mathbf{u}^h) = \int_{\Omega} \rho\, \mathbf{w}^h \cdot \mathbf{u}^h \nabla \mathbf{u}^h \, d\Omega$$

In the transient case the term is evaluated as follows:

$$\mathbf{N}(\mathbf{u}^h) = \int_{\Omega} \rho\, \mathbf{w}^h \cdot \mathbf{u}_{n+\beta}^h \mathbf{u}_{n+\alpha}^h \, d\Omega$$

where $\mathbf{u}_{n+\beta}^h$ is defined in a similar fashion as $\mathbf{u}_{n+\alpha}^h$ and can be expressed both in term of $\mathbf{u}_n^h$ and $\mathbf{u}_{n+1}^h$, i.e.

$$\mathbf{u}_{n+\beta}^h = \beta \mathbf{u}_{n+1}^h + (1-\beta)\mathbf{u}_n^h \tag{5.13}$$

$$\mathbf{u}_{n+\beta}^h = \frac{\beta}{\alpha}\mathbf{u}_{n+\alpha}^h + (1-\frac{\beta}{\alpha})\mathbf{u}_n^h \tag{5.14}$$

In the case of a steady state solution the $\mathbf{N}(\mathbf{u}^h)$ term has the following form

$$\mathbf{N}(\mathbf{u}^h) = \int_{\Omega} \rho\, \mathbf{w}^h \cdot \mathbf{u}_{n+1}^h \nabla \mathbf{u}_{n+1}^h \, d\Omega$$

which makes it highly non-linear.

By using the definitions for $\mathbf{u}^h$ and $\mathbf{w}^h$ from the previous section the term $\mathbf{N}(\mathbf{u}^h)$ can be evaluated in the following way by taking into account that the term is

not constant and that it needs to be integrated in three Gauss points. The weighting at each point is equal to $\frac{1}{3}$ in 2D and in 3D it is equal to $\frac{1}{4}$,

$$\begin{aligned}\mathbf{N}(\mathbf{u}^h) &\rightarrow \int_\Omega \rho\, \mathrm{N_A}\mathbf{w_A}\cdot(\mathrm{N_B}\mathbf{u}_\mathrm{B}^{n+\beta})\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\ &= \mathbf{w_A}\cdot\int_\Omega \rho\, \mathrm{N_A}(\mathrm{N_B}\mathbf{u}_\mathrm{B}^{n+\beta})\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\ &= \mathbf{w_A}\cdot\frac{\rho\, Ad}{3}\sum_{gp=1}^{NGauss}\sum_{\mathrm{B}=1}^{NNodes}\mathrm{N_A^{gp}N_B^{gp}}\mathbf{u}_\mathrm{B}^{n+\beta}\nabla\mathbf{u}^{n+\alpha}\end{aligned}$$

To make the code more efficient as in the previous subsection, the constant parts can be evaluated before hand and then used without the need to recalculate it again. First a constant $C^{\mathbf{N(u)}}$ that is the same for all nodes and all Gauss points can be defined in the following way,

	2D	3D
$C^{\mathbf{N(u)}}$	$\frac{\rho Ad}{3}$	$\frac{\rho V}{4}$

Table 5.2: Values for $C^{\mathbf{N(u)}}$

Secondly the term $\mathrm{N_B^{gp}}\mathbf{u}_\mathrm{B}^{n+\beta}\nabla\mathbf{u}^{n+\alpha}$ is constant for all nodes and can be calculated separately for every Gauss point yielding

$$\mathbf{UGradU}^{gp} = \sum_{\mathrm{B}=1}^{NNodes}\mathrm{N_B^{gp}}\mathbf{u}_\mathrm{B}^{n+\beta}\nabla\mathbf{u}^{n+\alpha} \tag{5.15}$$

The value of the force vector can be written as follows

$$\mathbf{F}_\mathrm{A}^{\mathbf{N(u)}} = C^{\mathbf{N(u)}}\sum_{gp=1}^{NGauss}\mathrm{N_A^{gp}}\mathbf{UGradU}^{gp} \tag{5.16}$$

5.2.3 Internal Force - Part $\mathbf{N}_\delta(\mathbf{u}^h)$

This is the part that represents streamline upwind contribution to the convective part of the problem,

$$\mathbf{N}_\delta(\mathbf{u}^h) = \int_\Omega \rho\tau(\mathbf{u}^h)(\mathbf{u}^h\cdot\nabla)\mathbf{w}^h\cdot\mathbf{u}^h\nabla\mathbf{u}^h\,\mathrm{d}\Omega$$

This term is non-linear and needs to be integrated in several Gauss points as with the term $\mathbf{N(u)}$. The part that is new and needs to be elaborated in some detail

is the value of the stabilising parameter $\tau(\mathbf{u}^h)$. The stabilising parameter $\tau(\mathbf{u}^h)$ is assumed to be constant within an element. The reason for this is that τ is an element parameter and so intuitively it makes sense to keep it constant throughout the element. The value for τ is defined via the element Reynolds number Re, the element Courant number $C_{\Delta t}$, the element length h_{len} and the velocity norm $\|\mathbf{u}\|$. The element Courant number is used only for transient analysis. The values for the various parameters were already mentioned in the previous chapter but will be repeated here in more detail. They are defined as follows:

$$h_{len} = \frac{A}{\pi} \quad \text{for 2D and} \quad h_{len} = \sqrt[3]{6V} \quad \text{for 3D}$$

$$Re = \frac{\|\mathbf{u}\| h_{len} \rho}{2\nu}, \quad \rho \text{ is the density and } \nu \text{ is the current viscosity}$$

$$C_{\Delta t} = \frac{\|\mathbf{u}\| \Delta t}{h_{len}}, \quad \Delta t \text{ is the time step}$$

$$z(Re, C_{\Delta t}) = \left[\left(\frac{1}{C_{\Delta t}} \right)^2 + 1 + \left(\frac{3}{Re} \right)^2 \right]^{-\frac{1}{2}}$$

$$\tau = \frac{h_{len}}{2\|\mathbf{u}\|} z(Re, C_{\Delta t})$$

To calculate τ the velocity at the centre of the element is chosen. A separate time fraction parameter γ is chosen similar to parameters α and β to calculate the velocity. The additional parameter has been added so that the time stepping scheme can be optimised.

$$\mathbf{u}^h_{n+\gamma} = \mathbf{u}^h_{n+1} + (1-\gamma)\mathbf{u}^h_n \tag{5.17}$$

$$\mathbf{u}^h_{n+\gamma} = \frac{\gamma}{\alpha}\mathbf{u}^h_{n+\alpha} + (1-\frac{\gamma}{\alpha})\mathbf{u}^h_n \tag{5.18}$$

From the tests that have been done in this work it has been observed that the best results are obtained when γ is equal to β. With τ defined at the centre of the element it is important to note that the velocity used in the stabilisation weighting function δ^h is also calculated at the centre and at the time instant $n+\gamma$. The stabilisation weighting function is defined as follows,

$$\delta^h = \tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma} \cdot \nabla)\mathbf{w}^h,$$

where c indicates that the velocity is calculated at the centre of the element. $\mathbf{N}_\delta(\mathbf{u})$ can now be evaluated in the following way

$$\begin{aligned}
\mathbf{N}_\delta(\mathbf{u}) &\rightarrow \int_\Omega \rho\tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma}\cdot\nabla)\mathrm{N_A}\mathbf{w_A}\cdot\mathrm{N_B}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\
&= \rho\int_\Omega \tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma}\cdot\nabla\mathrm{N_A})\mathbf{w_A}\cdot\mathrm{N_B}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\
&= \mathbf{w_A}\cdot\rho\int_\Omega \tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma}\cdot\nabla\mathrm{N_A})\mathrm{N_B}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\
&= \mathbf{w_A}\cdot\frac{\rho\,Ad}{3}\tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma}\cdot\nabla\mathrm{N_A})\sum_{gp=1}^{NGauss}\sum_{\mathrm{B}=1}^{NNodes}\mathrm{N}^{gp}_{\mathrm{B}}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}
\end{aligned}$$

From here the force vector can be written using $C^{\mathbf{N(u)}}$ and $\mathbf{UGradU}^{gp}$ as for the term $\mathbf{N(u)}$

$$\mathbf{F}^{\mathbf{N}_\delta}_{\mathrm{A}} = C^{\mathbf{N(u)}}\tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma}\cdot\nabla\mathrm{N_A})\sum_{gp=1}^{NGauss}\mathbf{UGradU}^{gp} \tag{5.19}$$

5.2.4 Internal Force - Part $\mathbf{N}_\epsilon$

This is the part that represents the pressure stabilisation contribution to the convective part of the problem,

$$\mathbf{N}_\epsilon(\mathbf{u}) = \int_\Omega \rho\tau(\mathbf{u}^h)\frac{1}{\rho}\nabla q^h\cdot\mathbf{u}^h\nabla\mathbf{u}^h\,\mathrm{d}\Omega = \int_\Omega \tau(\mathbf{u}^h)\nabla q^h\cdot\mathbf{u}^h\nabla\mathbf{u}^h\,\mathrm{d}\Omega$$

In an analogous way to $\mathbf{N}_\delta(\mathbf{u})$ this term consists of a stabilisation weighting function, in this case ϵ^h, and the convective term $\mathbf{N}$. The weighting function ϵ^h is defined as follows

$$\epsilon^h = \tau(\mathbf{u}^h)\frac{1}{\rho}\nabla q^h$$

The term $\mathbf{N}_\epsilon(\mathbf{u})$ is evaluated in a similar fashion as with the previous terms, i.e.

$$\begin{aligned}
\mathbf{N}_\epsilon(\mathbf{u}) &\rightarrow \int_\Omega \tau(\mathbf{u}^c_{n+\gamma})\nabla(\mathrm{N_A}q_\mathrm{A})\cdot\mathrm{N_B}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\
&= q_\mathrm{A}\int_\Omega \tau(\mathbf{u}^c_{n+\gamma})\nabla\mathrm{N_A}\cdot\mathrm{N_B}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}\,\mathrm{d}\Omega \\
&= q_\mathrm{A}\frac{Ad}{3}\tau(\mathbf{u}^c_{n+\gamma})\sum_{gp=1}^{NGauss}\sum_{\mathrm{B}=1}^{NNodes}\nabla\mathrm{N_A}\cdot\mathrm{N}^{gp}_{\mathrm{B}}\mathbf{u}^{n+\beta}_{\mathrm{B}}\nabla\mathbf{u}^{n+\alpha}
\end{aligned}$$

It is necessary to define a new constant $C^{\mathbf{N}_\epsilon(\mathbf{u})}$ see Table 5.3 With this new constant and $\mathbf{UGradU}^{gp}$ that has already been defined in (5.15) the force vector can be defined. It is important to note that E is used to indicate that this is a contribution to the forces that correspond to the pressure variables,

	2D	3D
$C^{\mathbf{N}_\epsilon(\mathbf{u})}$	$\frac{Ad}{3}$	$\frac{V}{4}$

Table 5.3: Values for $C^{\mathbf{N}_\epsilon(\mathbf{u})}$

$$\mathrm{E}_{\mathrm{A}}^{\mathbf{N}_\epsilon(\mathbf{u})} = C^{\mathbf{N}_\epsilon(\mathbf{u})}\tau(\mathbf{u}^c_{n+\gamma}) \sum_{gp=1}^{NGauss} \nabla \mathrm{N}_\mathrm{A} \cdot \mathbf{UGradU}^{gp} \tag{5.20}$$

5.2.5 Internal Force - Part $\mathbf{K(u)}$

The term $\mathbf{K(u)}$ is the viscous, dissipative part of the governing equation. This term represents the only term that is responsible for non-Newtonian behaviour. The non-Newtonian behaviour is exhibited through the non-linear dependence of the viscosity μ on the velocity gradient. The velocities are linear through the element which means that the velocity gradients are constant which in turn yield a constant viscosity for an element. It is sufficient to integrate the term in one Gauss point since the rest of the term involves strains which are also constant. The term $\mathbf{K(u)}$ is obtained as,

$$\mathbf{K(u)} = \int_\Omega 2\mu\, \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\mathbf{u}^h)\, \mathrm{d}\Omega,$$

where for the transient case the viscosity is calculated using $\mathbf{u}^{n+\alpha}$ as the velocity so that $\mu = \mu(\mathbf{u}^{n+\alpha})$. After using the expressions for $\mathbf{u}^h$ and $\mathbf{w}^h$ as in the previous section and keeping in mind the expression for $\boldsymbol{\varepsilon}(\mathbf{x}) = \frac{1}{2}\left(\nabla\mathbf{x} + \nabla^{\mathrm{T}}\mathbf{x}\right)$ the following expressions can be written

$$\begin{aligned}\mathbf{K(u)} &= \int_\Omega 2\mu\, \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha})\, \mathrm{d}\Omega = \int_\Omega 2\mu\, \boldsymbol{\varepsilon}(\mathrm{N_A}\mathbf{w_A}) : \boldsymbol{\varepsilon}(\sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}_\mathrm{B}^{n+\alpha})\, \mathrm{d}\Omega \\ &= \int_\Omega 2\mu \sum_{\mathrm{B}=1}^{NNodes} 0.5\,(\mathbf{w_A} \otimes \nabla\mathrm{N_A} + \nabla\mathrm{N_A} \otimes \mathbf{w_A}) : \\ &\quad 0.5\,(\mathbf{u}_\mathrm{B}^{n+\alpha} \otimes \nabla\mathrm{N_B} + \nabla\mathrm{N_B} \otimes \mathbf{u}_\mathrm{B}^{n+\alpha})\, \mathrm{d}\Omega\end{aligned}$$

leading to

$$\mathbf{K(u)} \rightarrow \int_{\Omega} 0.5\mu \sum_{B=1}^{NNodes} \Big[\mathbf{w}_A \cdot (\mathbf{u}_B^{n+\alpha} \otimes \nabla N_B)\nabla N_A + \mathbf{w}_A \cdot (\nabla N_B \otimes \mathbf{u}_B^{n+\alpha})\nabla N_A +$$
$$\mathbf{w}_A \cdot (\mathbf{u}_B^{n+\alpha} \otimes \nabla N_B)^T \nabla N_A + \mathbf{w}_A \cdot (\nabla N_B \otimes \mathbf{u}_B^{n+\alpha})^T \nabla N_A \Big] \, d\Omega$$
$$= \mathbf{w}_A \cdot \int_{\Omega} 0.5\mu \sum_{B=1}^{NNodes} 2(\mathbf{u}_B^{n+\alpha} \otimes \nabla N_B + \nabla N_B \otimes \mathbf{u}_B^{n+\alpha})\nabla N_A \, d\Omega$$
$$= \mathbf{w}_A \cdot Ad\mu \sum_{B=1}^{NNodes} (\mathbf{u}_B^{n+\alpha} \otimes \nabla N_B + \nabla N_B \otimes \mathbf{u}_B^{n+\alpha})\nabla N_A$$

As in the previous sections a constant $C^{\mathbf{K(u)}}$ can be defined to take into account the differences between 2D and 3D expressions, as follows

	2D	3D
$C^{\mathbf{K(u)}}$	$Ad\mu$	$V\mu$

Table 5.4: Values for $C^{\mathbf{K(u)}}$

The force vector for part $\mathbf{K(u)}$ finally takes the following form

$$\mathbf{F}_A^{\mathbf{K(u)}} = C^{\mathbf{K(u)}} \sum_{B=1}^{NNodes} (\mathbf{u}_B^{n+\alpha} \otimes \nabla N_B + \nabla N_B \otimes \mathbf{u}_B^{n+\alpha})\nabla N_A. \tag{5.21}$$

5.2.6 Internal Force - Part $\mathbf{G}_p$

The stress tensor is split into two parts (4.6); the first is the deviatoric stress component and the second is the pressure part. The term that corresponds to the pressure part is written as

$$\mathbf{G}_p = -\int_{\Omega} p^h \, \boldsymbol{\varepsilon}(\mathbf{w}^h) : \mathbf{I} \, d\Omega \tag{5.22}$$

The value for pressure is calculated at time $n + \alpha$ and can be written in a similar fashion as for the velocity

$$p^h = \sum_{B=1}^{NNodes} N_B p_B^{n+\alpha} \tag{5.23}$$

Following the procedure described earlier for the previous terms the expression for the term $\mathbf{G}_p$ can be obtained. It is important to note that the only non-constant term is the value for $\mathrm{N_B}$ which means that the integral can evaluated by using (5.10).

$$\begin{aligned}
\mathbf{G}_p \rightarrow & -\int_\Omega \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} p_\mathrm{B}^{n+\alpha}\, 0.5\, (\mathbf{w}_\mathrm{A} \otimes \nabla \mathrm{N_A} + \nabla \mathrm{N_A} \otimes \mathbf{w}_\mathrm{A}) : \mathbf{I}\, \mathrm{d}\Omega \\
= & -\int_\Omega \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} p_\mathrm{B}^{n+\alpha}\, 0.5 \cdot 2\, (\mathbf{w}_\mathrm{A} \cdot \nabla \mathrm{N_A})\, \mathrm{d}\Omega \\
= & \mathbf{w}_\mathrm{A} \cdot \left[-\int_\Omega \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} p_\mathrm{B}^{n+\alpha}\, \nabla \mathrm{N_A}\, \mathrm{d}\Omega \right] = \mathbf{w}_\mathrm{A} \cdot \left[-\frac{Ad}{3} \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_A} p_\mathrm{B}^{n+\alpha} \right]
\end{aligned}$$

Already defined constant $C^{\mathbf{N}_\epsilon(\mathbf{u})}$ can be used evaluate the force making this force equal to

$$\mathrm{E}_\mathrm{A}^{\mathbf{G}_p} = -C^{\mathbf{N}_\epsilon(\mathbf{u})} \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_A} p_\mathrm{B}^{n+\alpha}. \tag{5.24}$$

5.2.7 Internal Force - Part $\mathbf{G}_\delta$

This is the streamline upwind contribution to the pressure part of the stress term in the governing equations,

$$\mathbf{G}_\delta = \int_\Omega \tau(\mathbf{u}^h)(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot (\nabla \cdot p^h\, \mathbf{I})\, \mathrm{d}\Omega$$

The streamline upwind contribution is evaluated in the same way as with the term $\mathbf{N}_\delta$. The pressure is evaluated in the same way as for the term $\nabla \mathbf{p}$. Thus

$$\begin{aligned}
\mathbf{G}_\delta(\mathbf{u}, p) \rightarrow & \int_\Omega \tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma} \cdot \nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A} \cdot \nabla \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} p_\mathrm{B}^{n+\alpha}\, \mathrm{d}\Omega \\
= & \mathbf{w}_\mathrm{A} \cdot \int_\Omega \tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma} \cdot \nabla \mathrm{N_A}) \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_B} p_\mathrm{B}^{n+\alpha}\, \mathrm{d}\Omega \\
= & \mathbf{w}_\mathrm{A} \cdot Ad\tau(\mathbf{u}^c_{n+\gamma})(\mathbf{u}^c_{n+\gamma} \cdot \nabla \mathrm{N_A}) \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_B} p_\mathrm{B}^{n+\alpha}
\end{aligned}$$

The constant $C^{\mathbf{G}_\delta}$ is defined to be equal to the volume of Ω, which means that it is equal to Ad in the 2D case and V in the 3D case. The force vector can be

written as

$$\mathbf{F}_{\mathrm{A}}^{\mathbf{G}_\delta(\mathbf{u},p)} = C^{\mathbf{G}_\delta}\tau(\mathbf{u}_{n+\gamma}^c)(\mathbf{u}_{n+\gamma}^c \cdot \nabla \mathrm{N_A}) \sum_{gp=1}^{NGauss} \mathbf{UGradU}^{gp} \tag{5.25}$$

5.2.8 Internal Force - Part $\mathbf{G}_\epsilon$

This is the part that represents the pressure stabilisation contribution to the pressure term,

$$\mathbf{G}_\epsilon = \int_\Omega \tau \frac{1}{\rho} \nabla q^h \cdot (\nabla \cdot p^h \, \mathbf{I}) \, \mathrm{d}\Omega$$

Using the already established expressions the term $\mathbf{G}_\epsilon(\mathbf{u}, p)$ is evaluated in the following way,

$$\begin{aligned}
\mathbf{G}_\epsilon(\mathbf{u}, p) &\rightarrow \frac{1}{\rho}\int_\Omega \tau(\mathbf{u}_{n+\gamma}^c)\nabla(\mathrm{N_A q_A}) \cdot \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} p_{\mathrm{B}}^{n+\alpha} \, \mathrm{d}\Omega \\
&= \frac{1}{\rho}\int_\Omega \tau(\mathbf{u}_{n+\gamma}^c)\nabla \mathrm{N_A q_A} \cdot \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} p_{\mathrm{B}}^{n+\alpha} \, \mathrm{d}\Omega \\
&= \mathrm{q_A} \frac{1}{\rho}\int_\Omega \tau(\mathbf{u}_{n+\gamma}^c) \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_A} \cdot \mathrm{N_B} p_{\mathrm{B}}^{n+\alpha} \, \mathrm{d}\Omega \\
&= \mathrm{q_A} \frac{Ad}{\rho}\tau(\mathbf{u}_{n+\gamma}^c) \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_A} \cdot \mathrm{N_B} p_{\mathrm{B}}^{n+\alpha}
\end{aligned}$$

A new constant $C^{\mathbf{G}_\epsilon}$ is introduced and is given in Table 5.2.8

	2D	3D
$C^{\mathbf{G}_\epsilon}$	$\frac{Ad}{\rho}$	$\frac{V}{\rho}$

Table 5.5: Values for $C^{\mathbf{G}_\epsilon}$

The force can now be written as follows

$$\mathrm{E}_{\mathrm{A}}^{\mathbf{G}_\epsilon} = C^{\mathbf{G}_\epsilon}\tau(\mathbf{u}_{n+\gamma}^c) \sum_{\mathrm{B}=1}^{NNodes} \nabla \mathrm{N_A} \cdot \mathrm{N_B} p_{\mathrm{B}}^{n+\alpha} \tag{5.26}$$

5.2.9 Internal Force - Part $\mathbf{G}^{\mathrm{T}}$

This term represents the incompressibility condition,

$$\mathbf{G}^{\mathrm{T}}\mathbf{u} = \int_{\Omega} q^h \, \nabla \cdot \mathbf{u}^h \, \mathrm{d}\Omega$$

The term is evaluated as follows

$$\begin{aligned}
\mathbf{G}^{\mathrm{T}}\mathbf{u} &\rightarrow \int_{\Omega} \mathrm{q_A N_A} \nabla \cdot \left(\sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B} \mathbf{u}_{\mathrm{B}}^{n+\alpha} \right) \mathrm{d}\Omega \\
&= \mathrm{q_A} \int_{\Omega} \mathrm{N_A} \sum_{\mathrm{B}=1}^{NNodes} [\nabla \mathrm{N_B} \cdot \mathbf{u}_{\mathrm{B}}^{n+\alpha} + \mathrm{N_B}(\nabla \cdot \mathbf{u}_{\mathrm{B}}^{n+\alpha})] \, \mathrm{d}\Omega \qquad (\nabla \cdot \mathbf{u}_{\mathrm{B}}^{n+\alpha} = 0) \\
&= \mathrm{q_A} \frac{Ad}{3} \sum_{\mathrm{B}=1}^{NNodes} (\nabla \mathrm{N_B} \cdot \mathbf{u}_{\mathrm{B}}^{n+\alpha})
\end{aligned}$$

The constant is the same as for $\mathbf{N}_\epsilon$ so that

$$\mathrm{E}_{\mathrm{A}}^{\mathbf{G}^{\mathrm{T}}} = C^{\mathbf{N}_\epsilon(\mathbf{u})} \sum_{\mathrm{B}=1}^{NNodes} (\nabla \mathrm{N_B} \cdot \mathbf{u}_{\mathrm{B}}^{n+\alpha}) \tag{5.27}$$

5.3 Discretization of external forces

The external forces are calculated in a similar manner as the internal forces. In treating the external forces we distinguish two types. The first are body forces that act over elements. The second are traction forces on the element boundaries.

5.3.1 External force - Part F

This external force consists of the traction force prescribed on the boundary and body forces. The traction force is integrated on the boundary Γ_h where it is prescribed. The domain Γ_h is taken to be equal to the discretized domain. This type of approximation does introduce an error when the element boundaries do not exactly coincide with the original boundary, details can be found in Strang & Fix (1973). The body force is integrated over the domain of interest. The force vector has the following form

$$\mathbf{F} = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^{n+\alpha} \, d\Gamma + \int_{\Omega} \mathbf{w}^h \cdot \mathbf{f}^{n+\alpha} \, \mathrm{d}\Omega \tag{5.28}$$

where $\mathbf{h}^{n+\alpha}$ represents traction force and $\mathbf{f}^{n+\alpha}$ represents the body force. Both these values can be approximated by a linear interpolation over each element. This greatly facilitates the explicit calculation of forces. The linear interpolation introduces an $O(h^2)$ error (Davis, 1963).

5.3.2 External force - Part $\mathbf{F}_\delta$

This force represents the streamline upwind contribution of the body forces,

$$\mathbf{F}_\delta = \int_\Omega \tau(\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot \mathbf{f}^{n+\alpha}\, \mathrm{d}\Omega \tag{5.29}$$

5.3.3 External force - Part $\mathbf{E}_\epsilon$

This force represents the pressure stabilising contribution of the body forces.

$$\mathbf{E}_\epsilon = \int_\Omega \tau \frac{1}{\rho}\nabla q^h \cdot \mathbf{f}^{n+\alpha}\, \mathrm{d}\Omega \tag{5.30}$$

5.4 The Newton–Raphson solution method

Further to the time discretization the set of ordinary differential equations (4.35) of the previous chapter have been transformed into a set of non-linear algebraic equations, which now needs to be solved. According to the nature of the problem its solution may involve different procedures with corresponding different levels of complexity. In the simplest case the underlining problem is steady state in nature which implies that the equations need to be solved only once. Another case, is evolving steady state nature which requires some kind of evolution of the boundary conditions, or loadings, or both. In this case the evolution is split into discrete steps and within each step the problem is treated as steady state with boundary and loading conditions given by means of a defined value of the loading function. As a result we obtain a different set of equations at every step, which are often referred to as pseudo time-steps. Finally the most complex case involves a transient type problem where the governing equations need to be solved for every time step. In this case time dependent terms are also included.

Given the high non-linearity of the equations within every time step the equations are generally solved by means of an iterative procedure. As it has already been mentioned the Newton-Raphson procedure will be adopted. The noteworthy property of this iterative process is its quadratic convergence provided the solution estimate falls in an appropriate neighbourhood of the solution. This space in the current literature is often referred to as the ball of convergence. Before the Newton-Raphson solution procedure is described the concept of quadratic convergence will be clarified.

5.4.1 Convergence properties

The rate of convergence is used as a measure of the quality of an iterative solution procedure. The rate of convergence is assessed by comparing the residuals of two subsequent iterations. The residual is a global non negative measure which is related

to the error associated with the current iteration. In order to accept an iteration as the solution its residual has to be below a specified value, which is known as the residual tolerance.

Within a usual solution procedure, once the solution estimate falls inside the ball of convergence, the residual will gradually drop until it falls below the residual tolerance. If the residual between two iterations drops quadratically, the solution procedure is said to have quadratic convergence. This can be mathematically expressed as follows

$$\lim_{k\to\infty} \frac{R_{k+1}}{R_k^2} = \beta \geq 0$$

where R_{k+1} and R_k are residuals at iterations k and $k+1$ and β is a constant.

5.4.2 Directional Derivative

To be able to solve a set of non-linear algebraic equations using the Newton-Raphson procedure the equations need to be linearised. An important concept that is used in the linearization of the governing equations is the directional derivative. The directional derivative is a generalisation of the classic notion of derivative of a scalar function of a scalar variable.

To further elaborate the directional derivative concept the general function that is observed will be denoted by $\mathcal{F}$ whereas the set of variables will be denoted by $\mathbf{x}$. For instance $\mathcal{F}$ can represent a system of nonlinear equations, a matrix or a function, whereas $\mathbf{x}$ can be a list of variables or functions.

The set of non-linear algebraic equations that need to be solved via the Newton-Raphson procedure can now be expressed as

$$\mathcal{F}(\mathbf{x}) = \mathbf{0}. \tag{5.31}$$

Before the Newton-Raphson procedure is applied to this set of equations an overview of how this procedure is used to solve a single non-linear scalar equation will be illustrated. This will be then expanded to deal with solving sets of non-linear algebraic equations.

Newton-Raphson procedure for a scalar equations

This is the classical Newton-Raphson procedure. More details can be found in Stoer & Bulirsch (1980); Ortega & Rheinboldt (2000). The Newton-Raphson procedure as most other iterative procedures tries to determine a zero ξ of a scalar function f. The procedure starts at a point x_0, which is followed by successive approximations $x_i, i = 1, 2, \ldots$ that are calculated via an iteration function Φ that is defined in the following way:

$$x_{i+1} = \Phi(x_i), \qquad i = 0, 1, 2, \ldots$$

If these successive approximations lead to $\boldsymbol{\xi}$ ($\Phi(\boldsymbol{\xi}) = \boldsymbol{\xi}$) then $\boldsymbol{\xi}$ is a fixed point of Φ. The iteration function Φ is chosen in such a way that all fixed points of Φ are also zeros of f.

In order to apply the iterative procedure an iteration function has to be found. It can be found in a variety of ways. A systematic way for finding an iteration function Φ for a scalar function $f : \mathbb{R} \rightarrow \mathbb{R}$ that is sufficiently differentiable in a neighbourhood $\mathcal{N}(\xi)$ of ξ is by performing a Taylor series expansion of f about $x_0 \in \mathcal{N}(\xi)$. This yields

$$f(\xi) = 0 = f(x_0) + (\xi - x_0)f'(x_0) + \frac{(\xi - x_0)^2}{2!}f''(x_0) + \cdots \\ \frac{(\xi - x_0)^k}{k!}f^{(k)}(x_0 + \theta(\xi - x_0)) \quad 0 < \theta < 1$$

If the first two terms are retained as an approximation of f the following is obtained

$$0 = f(x_0) + (\xi - x_0)f'(x_0)$$

This in turn can be used to provide an approximation for ξ

$$\xi = x_0 - \frac{f(x_0)}{f'(x_0)}$$

If the value that is obtained for ξ is not satisfactory, it can be further improved by using the scheme from which it was derived, thus leading to the following iterative scheme

$$x_i = \Phi(x_{i+1}), \qquad \Phi(x) = x - \frac{f(x)}{f'(x)}$$

Newton-Raphson procedure for a set of non-linear equations

To be able to determine the iteration function $\boldsymbol{\Phi}$ for the problem (5.31) an additional variable ζ is introduced. This parameter enables us to define a new function $\mathbf{F}$ that depends solely on ζ. The new function represents the behaviour of the original function at a point $\mathbf{x}_0$ in the direction $\mathbf{u}$,

$$\mathbf{F}(\zeta) = \mathcal{F}(\mathbf{x}_0 + \zeta\mathbf{u}) \tag{5.32}$$

This is now a scalar function and a Taylor series expansion similar to the scalar case can be obtained. The expansion is done about $\zeta = 0$, which corresponds to $\mathbf{x} = \mathbf{x}_0$ and can be obtained as follows

$$\mathbf{F}(\zeta) = \mathbf{F}(0) + \zeta\frac{d\mathbf{F}}{d\zeta}\bigg|_{\zeta=0} + \frac{\zeta^2}{2!}\frac{d^2\mathbf{F}}{d\zeta^2}\bigg|_{\zeta=0} + \cdots$$

In terms of $\mathcal{F}$ this can be written as

$$\mathcal{F}(\mathbf{x}_0 + \zeta\mathbf{u}) = \mathcal{F}(\mathbf{x}_0) + \zeta\frac{d}{d\zeta}\bigg|_{\zeta=0}\mathcal{F}(\mathbf{x}_0 + \zeta\mathbf{u}) + \frac{\zeta^2}{2!}\frac{d^2}{d\zeta^2}\bigg|_{\zeta=0}\mathcal{F}(\mathbf{x}_0 + \zeta\mathbf{u}) + \cdots$$

By truncating this Taylor's series and by setting the variable $\zeta = 1$ the following linear approximation to the increment of $\mathcal{F}(\mathbf{x})$ is obtained

$$\mathcal{F}(\mathbf{x}_0 + \mathbf{u}) - \mathcal{F}(\mathbf{x}_0) \approx \frac{d}{d\zeta}\bigg|_{\zeta=0}\mathcal{F}(\mathbf{x}_0 + \zeta\mathbf{u}) \tag{5.33}$$

where the term on the right hand side is called a *directional derivative* of $\mathcal{F}(\mathbf{x})$ at $\mathbf{x}_0$ in the direction of $\mathbf{u}$ and has the following notation

$$\mathcal{D}\mathcal{F}(\mathbf{x}_0)[\mathbf{u}] = \frac{d}{d\zeta}\bigg|_{\zeta=0}\mathcal{F}(\mathbf{x}_0 + \zeta\mathbf{u}). \tag{5.34}$$

By using the above notation equation (5.33) can be written as

$$\mathcal{F}(\mathbf{x}_0 + \mathbf{u}) \approx \mathcal{F}(\mathbf{x}_0) + \mathcal{D}\mathcal{F}(\mathbf{x}_0)[\mathbf{u}] \tag{5.35}$$

By using the initial condition (5.31) and the above equation, the following equation is obtained

$$\mathcal{F}(\mathbf{x}_0) + \mathcal{D}\mathcal{F}(\mathbf{x}_0)[\mathbf{u}] = \mathbf{0}$$

This then in turn enables us to obtain the following general Newton-Raphson procedure

$$\mathcal{D}\mathcal{F}(\mathbf{x}_k)[\mathbf{u}] = -\mathcal{F}(\mathbf{x}_k) \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{u} \tag{5.36}$$

where $\mathcal{F}$ represents the governing equations that where derived in the previous section 5.2. More details about directional derivatives can be found in Bonet & Wood (1997).

5.5 Derivation of Stiffness Matrix

To be able to apply the Newton-Raphson procedure to governing equations (5.36) a directional derivative of the governing equations is necessary. This will be performed in turn for each term of the residual vector described in Section 5.2. As with discretization the various terms have similarities. These similarities mean that procedures have to be repeated for terms in question. Such procedures will be considered in detail the first time they occur. In subsequent cases references will be used. The derivation will be done for the transient case. To obtain the steady state forms the time parameters α, β and γ need to be set to 1 and the time dependent part of the governing equations needs to be dropped.

5.5.1 Derivation of the Stiffness matrix - Part Ma

This is the time dependent term of the governing equations. The directional derivative is taken with respect to the unknown nodal values $\mathbf{u}_B^{n+1}$ and p_B^{n+1} in the direction $\Delta\mathbf{u}_B$ and Δp_B. The term has the following form

$$\mathbf{Ma} = \int_{\Omega_e} \rho\, \mathbf{w}^h \cdot \frac{\partial \mathbf{u}^h}{\partial t}\, d\Omega$$

After substituting the values for $\mathbf{w}^h$ and $\mathbf{u}^h$ the following is obtained

$$\begin{aligned}
\mathbf{Ma} \rightarrow & \int_\Omega \rho\, \mathrm{N_A}\mathbf{w}_A \cdot \mathrm{N_B} \frac{\partial \mathbf{u}_B}{\partial t}\, \mathrm{d}\Omega \\
&= \mathbf{w}_A \cdot \int_\Omega \rho\, \mathrm{N_A N_B} \frac{\mathbf{u}_B^{n+1} - \mathbf{u}_B^n}{\Delta t}\, \mathrm{d}\Omega = \mathbf{w}_A \cdot \frac{\rho}{\Delta t} \int_\Omega \mathrm{N_A N_B} \mathbf{I}\, \mathrm{d}\Omega (\mathbf{u}_B^{n+1} - \mathbf{u}_B^n) \\
&= \mathbf{w}_A \cdot \frac{\rho}{\Delta t} \mathbf{M}_{AB} (\mathbf{u}_B^{n+1} - \mathbf{u}_B^n)
\end{aligned}$$

where $\mathbf{M}_{AB}$ is defined in section 5.2.1. The directional derivative can now be written as follows

$$\begin{aligned}
\mathcal{D}\mathbf{Ma}[\Delta\mathbf{u}_B] &= \mathbf{w}_A \cdot \frac{\rho}{\Delta t} \mathbf{M}_{AB} \left.\frac{d}{d\zeta}\right|_{\zeta=0} [(\mathbf{u}_B^{n+1} + \zeta\, \Delta\mathbf{u}_B) - \mathbf{u}_B^n] \\
&= \mathbf{w}_A \cdot \frac{\rho}{\Delta t} \mathbf{M}_{AB}\, \Delta\mathbf{u}_B
\end{aligned}$$

5.5.2 Derivation of the Stiffness matrix - Part N(u)

This is the convective term and it is equal to

$$\mathbf{N}(\mathbf{u}) = \int_\Omega \rho\, \mathbf{w}^h \cdot \mathbf{u}_{n+\beta}^h \nabla \mathbf{u}_{n+\alpha}^h\, \mathrm{d}\Omega$$

It is important to note that the governing equations are solved for $\mathbf{u}_{n+1}^h$. This means that $\mathbf{u}_{n+\alpha}^h$ and $\mathbf{u}_{n+\beta}^h$ need to be treated as functions of $\mathbf{u}_{n+1}^h$. It can be easily shown that by using the definitions for $\mathbf{u}_{n+\alpha}^h$ (5.2) and $\mathbf{u}_{n+\beta}^h$ (5.13) that the directional derivatives in the direction of $\Delta\mathbf{u}^h$ can be written as

$$\mathcal{D}\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1})[\Delta\mathbf{u}^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} \mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h) = \frac{d}{d\zeta}\bigg|_{\zeta=0} \left[\alpha(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h) + (1-\alpha)\mathbf{u}^h_n\right] = \alpha\Delta\mathbf{u}^h \tag{5.37}$$

$$\mathcal{D}\mathbf{u}^h_{n+\beta}(\mathbf{u}^h_{n+1})[\Delta\mathbf{u}^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} \mathbf{u}^h_{n+\beta}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h) = \frac{d}{d\zeta}\bigg|_{\zeta=0} \left[\beta(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h) + (1-\beta)\mathbf{u}^h_n\right] = \beta\Delta\mathbf{u}^h \tag{5.38}$$

Unlike for the previous derivation for this term the derivation will be performed first in terms of $\Delta\mathbf{u}^h$ and then expanded in terms of $\Delta\mathbf{u}_\mathrm{B}$. We note that the increments $\Delta\mathbf{u}^h$ and $\Delta\mathbf{u}_\mathrm{B}$ are related via the shape function $\mathrm{N_B}$ as follows $\Delta\mathbf{u}^h = \mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}$. Thus

$$\begin{aligned}
\mathcal{D}\mathbf{N}[\Delta\mathbf{u}^h] &= \frac{d}{d\zeta}\bigg|_{\zeta=0} \int_\Omega \rho\, \mathbf{w}^h \cdot \mathbf{u}^h_{n+\beta}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\nabla\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\,\mathrm{d}\Omega = \\
&= \int_\Omega \rho\, \mathbf{w}^h \cdot \frac{d}{d\zeta}\bigg|_{\zeta=0} \mathbf{u}^h_{n+\beta}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\nabla\mathbf{u}^h_{n+\alpha}\,\mathrm{d}\Omega \\
&+ \int_\Omega \rho\, \mathbf{w}^h \cdot \mathbf{u}^h_{n+\beta}\frac{d}{d\zeta}\bigg|_{\zeta=0} \nabla\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\,\mathrm{d}\Omega \\
&= \int_\Omega \beta\rho\, \mathbf{w}^h \cdot \Delta\mathbf{u}^h\nabla\mathbf{u}^h_{n+\alpha}\,\mathrm{d}\Omega + \int_\Omega \alpha\rho\, \mathbf{w}^h \cdot \mathbf{u}^h_{n+\beta}\nabla(\Delta\mathbf{u}^h)\,\mathrm{d}\Omega
\end{aligned}$$

After replacing the weighting (test) function $\mathbf{w}^h$ (see (5.4)) and the trial function $\mathbf{u}^h$ (see (5.5),(5.6),(5.7)) in terms of their shape functions and nodal values, the values for $\nabla\mathbf{u}^h_{n+\alpha}$ and $\mathbf{u}^h_{n+\beta}$ are evaluated and give the following

$$\begin{aligned}
\nabla\mathbf{u}^h_{n+\alpha} &= \nabla\left(\sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}^{n+\alpha}_\mathrm{B}\right) = \\
&= \sum_{\mathrm{B}=1}^{NNodes} (\mathbf{u}^{n+\alpha}_\mathrm{B} \otimes \nabla\mathrm{N_B}) = \nabla\mathbf{u}_{n+\alpha} \\
\mathbf{u}^h_{n+\beta} &= \sum_{\mathrm{B}=1}^{NNodes} \mathrm{N_B}\mathbf{u}^{n+\beta}_\mathrm{B} = \mathbf{u}_{n+\beta}
\end{aligned}$$

The directional derivative can now be further evaluated as follows

$$\begin{aligned}\mathcal{D}\mathbf{N}(\Delta\mathbf{u}_\mathrm{B}) &\rightarrow \int_\Omega \rho\, \mathrm{N_A}\mathbf{w}_\mathrm{A} \cdot [\beta \mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha} + \alpha\mathbf{u}_{n+\beta}(\Delta\mathbf{u}_\mathrm{B} \otimes \nabla \mathrm{N_B})]\ \mathrm{d}\Omega \\ &= \int_\Omega \rho\, \mathrm{N_A}\mathbf{w}_\mathrm{A} \cdot \left[\beta \mathrm{N_B}\nabla\mathbf{u}^h_{n+\alpha}\Delta\mathbf{u}_\mathrm{B} + \alpha(\nabla \mathrm{N_B} \cdot \mathbf{u}^h_{n+\beta})\Delta\mathbf{u}_\mathrm{B}\right]\ \mathrm{d}\Omega \\ &= \mathbf{w}_\mathrm{A} \cdot \int_\Omega \rho\, \mathrm{N_A}\left[\beta \mathrm{N_B}\nabla\mathbf{u}^h_{n+\alpha} + \alpha(\nabla \mathrm{N_B} \cdot \mathbf{u}^h_{n+\beta})\mathbf{I}\right]\ \mathrm{d}\Omega\Delta\mathbf{u}_\mathrm{B}\end{aligned}$$

To obtain the final form the above equation needs to be integrated. The integration is performed in the same way as for the internal forces (3 Gauss points with $\frac{1}{3}$ weighting at each point in 2D and 4 Gauss points with $\frac{1}{4}$ weighting at each point in 3D). The term is as

$$\begin{aligned}\mathcal{D}\mathbf{N}(\Delta\mathbf{u}_\mathrm{B}) &\rightarrow \mathbf{w}_\mathrm{A} \cdot \frac{\rho\, A}{3} \sum_{gp=1}^{NGauss} \mathrm{N}^{gp}_\mathrm{A}\left[\beta \mathrm{N}^{gp}_\mathrm{B}\nabla\mathbf{u}^h_{n+\alpha} + \alpha(\nabla \mathrm{N_B} \cdot \mathbf{u}^{gp}_{n+\beta})\mathbf{I}\right]\Delta\mathbf{u}_\mathrm{B} \\ &= \mathbf{N}^1 + \mathbf{N}^2\end{aligned}$$

Superscript *gp* indicates that the corresponding terms change values at Gauss points.

5.5.3 Derivation of the Stiffness matrix - Part $\mathbf{N}_\delta(\mathbf{u})$

This is the term that represents the streamline upwind contribution to the convective part of the problem,

$$\mathbf{N}_\delta[\mathbf{u}^h] = \sum_{e=1}^{n_{el}} \int_{\Omega_e} \rho\tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma} \cdot \nabla)\mathbf{w}^h \cdot \mathbf{u}^h_{n+\beta}\nabla\mathbf{u}^h_{n+\alpha}\, \mathrm{d}\Omega$$

It is evaluated in much the same way as the previous term except that it also involves terms that are dependent on $\mathbf{u}^h_{n+\gamma}$, i.e.

$$\begin{aligned}\mathcal{D}\mathbf{N}_\delta(\mathbf{u}^h_{n+1})[\Delta\mathbf{u}^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} \int_{\Omega_e} \rho\tau[\mathbf{u}^h_{n+\gamma}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)] \\ [\mathbf{u}^h_{n+\gamma}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h) \cdot \nabla]\mathbf{w}^h \cdot \\ \mathbf{u}^h_{n+\beta}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\nabla\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\, \mathrm{d}\Omega \quad (5.39)\end{aligned}$$

The derivative of $\mathbf{u}^h_{n+\gamma}$ with respect to $\Delta\mathbf{u}^h$ can be obtained in a similar way as for $\mathbf{u}^h_{n+\alpha}$ and is equal to

$$\mathcal{D}\mathbf{u}^h_{n+\gamma}(\mathbf{u}^h_{n+1})[\Delta\mathbf{u}^h] = \gamma\Delta\mathbf{u}^h$$

It can be observed that equation (5.39) involves four terms that are dependent on $\mathbf{u}^h_{n+1}$ and this will lead to four different parts of the stiffness matrix after applying the product rule.

$$\begin{aligned}\mathcal{D}\mathbf{N}_\delta[\Delta\mathbf{u}^h] = \int_{\Omega_e} & \rho\frac{d}{d\zeta}\bigg|_{\zeta=0}[\tau(\mathbf{u}^h_{n+\gamma})](\mathbf{u}^h_{n+\gamma}\cdot\nabla)\mathbf{w}^h\cdot\mathbf{u}^h_{n+\beta}\nabla(\mathbf{u}^h_{n+\alpha}) \\ & + \rho\tau(\mathbf{u}^h_{n+\gamma})\frac{d}{d\zeta}\bigg|_{\zeta=0}[\mathbf{u}^h_{n+\gamma}\cdot\nabla]\mathbf{w}^h\cdot\mathbf{u}^h_{n+\beta}\nabla(\mathbf{u}^h_{n+\alpha}) \\ & + \rho\tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma}\cdot\nabla)\mathbf{w}^h\cdot\beta\Delta\mathbf{u}^h\nabla\mathbf{u}^h_{n+\alpha} \\ & + \rho\tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma}\cdot\nabla)\mathbf{w}^h\cdot\mathbf{u}^h_{n+\beta}\nabla(\alpha\Delta\mathbf{u}^h)\,\mathrm{d}\Omega\end{aligned}$$

The directional derivative of τ with respect to $\Delta\mathbf{u}^h$ is given in appendix A. The derivative is denoted by $\boldsymbol{\tau}'$ and is a vector. The derivative of $\mathbf{N}_\delta$ can now expressed as

$$\begin{aligned}\mathcal{D}\mathbf{N}_\delta[\Delta\mathbf{u}^h] = \int_{\Omega_e} & \rho[\boldsymbol{\tau}'(\mathbf{u}^h_{n+\gamma})\cdot\Delta\mathbf{u}^h](\mathbf{u}^h_{n+\gamma}\cdot\nabla)\mathbf{w}^h\cdot\mathbf{u}^h_{n+\beta}\nabla\mathbf{u}^h_{n+\alpha} \\ & + \rho\tau(\mathbf{u}^h_{n+\gamma})(\gamma\Delta\mathbf{u}^h\cdot\nabla)\mathbf{w}^h\cdot\mathbf{u}^h_{n+\beta}\nabla(\mathbf{u}^h_{n+\alpha}) \\ & + \rho\tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma}\cdot\nabla)\mathbf{w}^h\cdot\beta\Delta\mathbf{u}^h\nabla\mathbf{u}^h_{n+\alpha} \\ & + \rho\tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma}\cdot\nabla)\mathbf{w}^h\cdot\mathbf{u}^h_{n+\beta}\nabla(\alpha\Delta\mathbf{u}^h)\,\mathrm{d}\Omega\end{aligned}$$

When the shape functions are substituted the following is obtained

$$\begin{aligned}\mathcal{D}\mathbf{N}_\delta[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \rho\int_{\Omega_e} & (\boldsymbol{\tau}'_\mathrm{B}(\mathbf{u}_{n+\gamma})\cdot\Delta\mathbf{u}_\mathrm{B})(\mathbf{u}_{n+\gamma}\cdot\nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A}\cdot(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \\ & + \tau(\mathbf{u}_{n+\gamma})\mathrm{N_B}(\gamma\Delta\mathbf{u}_\mathrm{B}\cdot\nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A}\cdot(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \\ & + \tau(\mathbf{u}_{n+\gamma})(\mathbf{u}_{n+\gamma}\cdot\nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A}\cdot(\beta\mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha}) \\ & + \tau(\mathbf{u}_{n+\gamma})(\mathbf{u}_{n+\gamma}\cdot\nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A}\cdot[\mathbf{u}_{n+\beta}(\alpha\Delta\mathbf{u}_\mathrm{B}\otimes\nabla\mathrm{N_B})]\,\mathrm{d}\Omega\end{aligned}$$

For reasons of clarity the stabilisation parameter $\tau(\mathbf{u}_{n+\gamma})$ is denoted by τ, and the same is done for $\boldsymbol{\tau}'_\mathrm{B}(\mathbf{u}_{n+\gamma})$.

$$\begin{aligned}\mathcal{D}\mathbf{N}_\delta[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \rho\int_{\Omega_e} & (\boldsymbol{\tau}'_\mathrm{B}\cdot\Delta\mathbf{u}_\mathrm{B})(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N_A})\mathbf{w}_\mathrm{A}\cdot(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \\ & + \gamma\tau\mathrm{N_B}(\Delta\mathbf{u}_\mathrm{B}\cdot\nabla\mathrm{N_A})\mathbf{w}_\mathrm{A}\cdot(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \\ & + \beta\tau\mathrm{N_B}(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N_A})\mathbf{w}_\mathrm{A}\cdot(\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha}) \\ & + \alpha\tau(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N_A})\mathbf{w}_\mathrm{A}\cdot(\mathbf{u}_{n+\beta}\cdot\nabla\mathrm{N_B})\Delta\mathbf{u}_\mathrm{B}\,\mathrm{d}\Omega\end{aligned}$$

The vector $\mathbf{w}_A$ can be taken outside the integral leading to

$$\begin{aligned}\mathcal{D}\mathrm{N}_\delta[\Delta\mathbf{u}_B] \rightarrow \rho\mathbf{w}_A \cdot \int_{\Omega_e} \tau\left(\frac{\boldsymbol{\tau}'_B}{\tau}\cdot\Delta\mathbf{u}_B\right)(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N}_A)(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \\ + \gamma\tau\mathrm{N}_B(\Delta\mathbf{u}_B\cdot\nabla\mathrm{N}_A)(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \\ + \beta\tau\mathrm{N}_B(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N}_A)(\Delta\mathbf{u}_B\nabla\mathbf{u}_{n+\alpha}) \\ + \alpha\tau(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N}_A)(\mathbf{u}_{n+\beta}\cdot\nabla\mathrm{N}_B)\Delta\mathbf{u}_B\,\mathrm{d}\Omega.\end{aligned}$$

The same can be done with $\Delta\mathbf{u}_B$ after several manipulations

$$\begin{aligned}\mathcal{D}\mathrm{N}_\delta[\Delta\mathbf{u}_B] \rightarrow \rho\mathbf{w}_A \cdot \int_{\Omega_e} \tau(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N}_A)\left[(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha})\otimes\frac{\boldsymbol{\tau}'_B}{\tau}\right] \\ + \gamma\tau\mathrm{N}_B(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}\otimes\nabla\mathrm{N}_A) \\ + \beta\tau\mathrm{N}_B(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N}_A)\nabla\mathbf{u}_{n+\alpha} \\ + \alpha\tau(\mathbf{u}_{n+\gamma}\cdot\nabla\mathrm{N}_A)(\mathbf{u}_{n+\beta}\cdot\nabla\mathrm{N}_B)\mathbf{I}\,\mathrm{d}\Omega\Delta\mathbf{u}_B.\end{aligned}$$

The integration is performed in the same way as for $\mathbf{N}(\mathbf{u})$ giving finally

$$\begin{aligned}\mathcal{D}\mathrm{N}_\delta[\Delta\mathbf{u}_B] \rightarrow \mathbf{w}_A\cdot\left[\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\tau(\mathbf{u}^{gp}_{n+\gamma}\cdot\nabla\mathrm{N}_A)\left[(\mathbf{u}^{gp}_{n+\beta}\nabla\mathbf{u}_{n+\alpha})\otimes\frac{\boldsymbol{\tau}'^{gp}_B}{\tau}\right]\right]\Delta\mathbf{u}_B \\ + \mathbf{w}_A\cdot\left[\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\gamma\tau\mathrm{N}^{gp}_B(\mathbf{u}^{gp}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}\otimes\nabla\mathrm{N}_A)\right]\Delta\mathbf{u}_B \\ + \mathbf{w}_A\cdot\left[\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\beta\tau\mathrm{N}^{gp}_B(\mathbf{u}^{gp}_{n+\gamma}\cdot\nabla\mathrm{N}_A)\nabla\mathbf{u}_{n+\alpha}\right]\Delta\mathbf{u}_B \\ + \mathbf{w}_A\cdot\left[\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\alpha\tau(\mathbf{u}^{gp}_{n+\gamma}\cdot\nabla\mathrm{N}_A)(\mathbf{u}^{gp}_{n+\beta}\cdot\nabla\mathrm{N}_B)\mathbf{I}\right]\Delta\mathbf{u}_B \\ = \mathbf{N}^1_\delta + \mathbf{N}^2_\delta + \mathbf{N}^3_\delta + \mathbf{N}^4_\delta.\end{aligned}$$

5.5.4 Derivation of the Stiffness matrix - Part $\mathbf{N}_\epsilon$

This is the term that represents the pressure stabilisation contribution to the convective part of the problem, and takes the form

$$\mathbf{N}_\epsilon = \int_\Omega \rho\tau(\mathbf{u}^h)\frac{1}{\rho}\nabla\mathrm{q}^h\cdot\mathbf{u}^h\nabla\mathbf{u}^h\,\mathrm{d}\Omega = \int_\Omega \tau(\mathbf{u}^h)\nabla\mathrm{q}^h\cdot\mathbf{u}^h\nabla\mathbf{u}^h\,\mathrm{d}\Omega$$

As with the previous derivations the starting point is the definition for a directional derivative

$$\mathcal{D}\mathbf{N}_\epsilon[\Delta\mathbf{u}^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} \int_{\Omega_e} \tau[\mathbf{u}^h_{n+\gamma}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)]\nabla \mathrm{q}^h \cdot \mathbf{u}^h_{n+\beta}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h) \nabla\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1} + \zeta\Delta\mathbf{u}^h)\, \mathrm{d}\Omega$$

After using the product rule the following three terms are obtained

$$\begin{aligned}\mathcal{D}\mathbf{N}_\epsilon[\Delta\mathbf{u}^h] &= \int_{\Omega_e} (\boldsymbol{\tau}'(\mathbf{u}^h_{n+\gamma}) \cdot \Delta\mathbf{u}^h)\nabla \mathrm{q}^h \cdot \mathbf{u}^h_{n+\beta}\nabla\mathbf{u}^h_{n+\alpha} + \beta\tau(\mathbf{u}^h_{n+\gamma})\nabla \mathrm{q}^h \cdot \Delta\mathbf{u}^h\nabla\mathbf{u}^h_{n+\alpha} \\ &\quad + \alpha\tau(\mathbf{u}^h_{n+\gamma})\nabla \mathrm{q}^h \cdot \mathbf{u}^h_{n+\beta}\nabla(\Delta\mathbf{u}^h)\, \mathrm{d}\Omega \\ &= \int_{\Omega_e} \nabla \mathrm{q}^h \cdot \Big[(\boldsymbol{\tau}'(\mathbf{u}^h_{n+\gamma}) \cdot \Delta\mathbf{u}^h)\mathbf{u}^h_{n+\beta}\nabla\mathbf{u}^h_{n+\alpha} + \beta\tau(\mathbf{u}^h_{n+\gamma})\Delta\mathbf{u}^h\nabla\mathbf{u}^h_{n+\alpha} \\ &\quad + \alpha\tau(\mathbf{u}^h_{n+\gamma})\mathbf{u}^h_{n+\beta}\nabla(\Delta\mathbf{u}^h)\Big]\, \mathrm{d}\Omega\end{aligned}$$

Shape function substitution gives

$$\begin{aligned}\mathcal{D}\mathbf{N}_\epsilon[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \int_{\Omega_e} \nabla(\mathrm{N_A q_A}) \cdot \Big[&(\boldsymbol{\tau}'_\mathrm{B}(\mathbf{u}_{n+\gamma}) \cdot \Delta\mathbf{u}_\mathrm{B})\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha} \\ &+ \beta\tau(\mathbf{u}_{n+\gamma})\mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha} + \alpha\tau(\mathbf{u}_{n+\gamma})(\mathbf{u}_{n+\beta}\nabla(\mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}))\Big]\, \mathrm{d}\Omega\end{aligned}$$

Several manipulation are required in order rearrange the terms so that $\Delta\mathbf{u}_\mathrm{B}$ can be moved to the right hand side, which leads to

$$\begin{aligned}\mathcal{D}\mathbf{N}_\epsilon[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \int_{\Omega_e} \mathrm{q_A}\nabla\mathrm{N_A} \cdot \Big[&(\boldsymbol{\tau}'_\mathrm{B} \cdot \Delta\mathbf{u}_\mathrm{B})\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha} \\ &+ \beta\tau\mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha} + \alpha\tau\mathbf{u}_{n+\beta}(\Delta\mathbf{u}_\mathrm{B} \otimes \nabla\mathrm{N_B})\Big]\, \mathrm{d}\Omega\end{aligned}$$

$$\begin{aligned}\mathcal{D}\mathbf{N}_\epsilon[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \int_{\Omega_e} \mathrm{q_A}\nabla\mathrm{N_A} \cdot \Big\{&\Big[(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha}) \otimes \boldsymbol{\tau}'_\mathrm{B}\Big]\Delta\mathbf{u}_\mathrm{B} \\ &+ \beta\tau\mathrm{N_B}\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha} + \alpha\tau(\nabla\mathrm{N_B} \cdot \mathbf{u}_{n+\beta})\Delta\mathbf{u}_\mathrm{B}\Big\}\, \mathrm{d}\Omega\end{aligned}$$

$$\mathcal{D}\mathrm{N}_\epsilon[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \int_{\Omega_e} \mathrm{q}_\mathrm{A}\Big\{\nabla\mathrm{N}_\mathrm{A}\cdot\Big[(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha})\otimes\boldsymbol{\tau}'_\mathrm{B}\Big]\Delta\mathbf{u}_\mathrm{B} + \beta\tau\mathrm{N}_\mathrm{B}\nabla\mathrm{N}_\mathrm{A}\cdot\Delta\mathbf{u}_\mathrm{B}\nabla\mathbf{u}_{n+\alpha} + \alpha\tau(\nabla\mathrm{N}_\mathrm{B}\cdot\mathbf{u}_{n+\beta})\nabla\mathrm{N}_\mathrm{A}\cdot\Delta\mathbf{u}_\mathrm{B}\Big\}\,\mathrm{d}\Omega$$

$$\mathcal{D}\mathrm{N}_\epsilon[\Delta\mathbf{u}_\mathrm{B}] \rightarrow \mathrm{q}_\mathrm{A}\int_{\Omega_e}\Big\{\Big[\boldsymbol{\tau}'_\mathrm{B}\otimes(\mathbf{u}_{n+\beta}\nabla\mathbf{u}_{n+\alpha})\Big]\nabla\mathrm{N}_\mathrm{A} + \beta\tau\mathrm{N}_\mathrm{B}\nabla^T\mathbf{u}_{n+\alpha}\nabla\mathrm{N}_\mathrm{A} + \alpha\tau(\nabla\mathrm{N}_\mathrm{B}\cdot\mathbf{u}_{n+\beta})\nabla\mathrm{N}_\mathrm{A}\Big\}\,\mathrm{d}\Omega\cdot\Delta\mathbf{u}_\mathrm{B}$$

After the right form is obtained integration can be performed in a the same way as described earlier leading to

$$\begin{aligned}\mathcal{D}\mathrm{N}_\epsilon[\Delta\mathbf{u}_\mathrm{B}] \rightarrow\ & \mathrm{q}_\mathrm{A}\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\tau\left[\frac{\boldsymbol{\tau}'^{gp}_\mathrm{B}}{\tau}\otimes(\mathbf{u}^{gp}_{n+\beta}\nabla\mathbf{u}_{n+\alpha})\right]\nabla\mathrm{N}_\mathrm{A}\cdot\Delta\mathbf{u}_\mathrm{B}\\ & + \mathrm{q}_\mathrm{A}\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\beta\tau\mathrm{N}^{gp}_\mathrm{B}\nabla^T\mathbf{u}_{n+\alpha}\nabla\mathrm{N}_\mathrm{A}\cdot\Delta\mathbf{u}_\mathrm{B}\\ & + \mathrm{q}_\mathrm{A}\frac{\rho A}{3}\sum_{gp=1}^{NGauss}\tau(\nabla\mathrm{N}_\mathrm{B}\cdot\mathbf{u}^{gp}_{n+\beta})\nabla\mathrm{N}_\mathrm{A}\cdot\Delta\mathbf{u}_\mathrm{B}\\ & = \mathbf{N}^1_\epsilon + \mathbf{N}^2_\epsilon + \mathbf{N}^3_\epsilon\end{aligned}$$

5.5.5 Derivation of the Stiffness matrix - Part $\mathbf{K}(\mathbf{u})$

It has been already stated that this is the viscous part of the governing equation. It is highly non-linear as a consequence of the non-Newtonian character of the fluid considered. The successful linearization of this term is crucial in order to ensure a successful solution of the problems considered in this thesis. It is also important in accurately modelling non-Newtonian behaviour. The term under consideration can be written as

$$\mathbf{K}(\mathbf{u}) = \int_\Omega 2\mu\,\boldsymbol{\varepsilon}(\mathbf{w}^h)\,:\,\boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha})\,\mathrm{d}\Omega = \int_\Omega 2\mu(\mathbf{u}^h_{n+\alpha})\,\boldsymbol{\varepsilon}(\mathbf{w}^h)\,:\,\boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha})\,\mathrm{d}\Omega$$

Starting with the definition for a directional derivative the following is obtained

$$\mathcal{D}\mathbf{K}[\Delta\mathbf{u}^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} \int_\Omega 2\,\mu(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h))\,\boldsymbol{\varepsilon}(\mathbf{w}^h) \,:\, \boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h))\,\mathrm{d}\Omega =$$

$$= \int_\Omega 2\,\mu(\mathbf{u}^h_{n+\alpha})\,\boldsymbol{\varepsilon}(\mathbf{w}^h) \,:\, \frac{d\boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h))}{d\zeta}\bigg|_{\zeta=0} \mathrm{d}\Omega +$$

$$+ \int_\Omega 2\,\frac{d\mu(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h))}{d\zeta}\bigg|_{\zeta=0} \boldsymbol{\varepsilon}(\mathbf{w}^h) \,:\, \boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha})\,\mathrm{d}\Omega$$

The above derivation has two parts. The first part is the derivative of the strain tensor which yields the following

$$\frac{d\boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h))}{d\zeta}\bigg|_{\zeta=0} = \nabla^S\left(\frac{d}{d\zeta}\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h)\Big|_{\zeta=0}\right) =$$

$$\nabla^S(\alpha\Delta\mathbf{u}^h) = \alpha\nabla^S(\Delta\mathbf{u}^h) = \alpha\boldsymbol{\varepsilon}(\Delta\mathbf{u}^h)$$

The second term is the derivative of the viscosity μ, which can be obtained as

$$\frac{d\mu(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+h\Delta\mathbf{u}^h))}{d\zeta}\bigg|_{\zeta=0} =$$

$$\frac{d\mu}{d\dot\gamma}\frac{d\dot\gamma}{d\boldsymbol{\varepsilon}} : \frac{d\boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha}(\mathbf{u}^h_{n+1}+\zeta\Delta\mathbf{u}^h))}{d\zeta}\bigg|_{\zeta=0} = \alpha\frac{d\mu}{d\dot\gamma}\frac{d\dot\gamma}{d\boldsymbol{\varepsilon}} : \boldsymbol{\varepsilon}(\Delta\mathbf{u}^h)$$

After applying the chain rule it can be observed that the only expressions required to evaluate the above equation are derivatives $\frac{d\mu}{d\dot\gamma}$ and $\frac{d\dot\gamma}{d\boldsymbol{\varepsilon}}$. These will be now considered in turn:

$\frac{d\mu}{d\dot\gamma}$: This term is different for every fluid model. A general variable μ' will be used to denote this value. The following illustrates the values for μ' for different types of the non-Newtonian fluids:

- Newtonian Fluid

$$\mu(\dot\gamma) = \mu = \text{const}$$
$$\mu'(\dot\gamma) = 0$$

- Power Law Fluid

$$\mu(\dot\gamma) = K\,\dot\gamma^{n-1}$$
$$\mu'(\dot\gamma) = K\,(n-1)\,\dot\gamma^{n-2}$$

- Bird-Carreau Fluid

$$\mu(\dot{\gamma}) = \eta_\infty + (\eta_0 - \eta_\infty)\left[1 + (\lambda\dot{\gamma})^2\right]^{\frac{n-1}{2}}$$
$$\mu'(\dot{\gamma}) = \dot{\gamma}\,(\eta_0 - \eta_\infty)\,(n-1)\left[1 + (\lambda\dot{\gamma})^2\right]^{\frac{n-3}{2}}$$

- Bingham Fluid

$$\mu(\dot{\gamma}) = \frac{\sigma_Y}{\dot{\gamma}} + K$$
$$\mu'(\dot{\gamma}) = -\frac{\sigma_Y}{\dot{\gamma}^2}$$

- Herschel-Bulkley Fluid

$$\mu(\dot{\gamma}) = \frac{\sigma_Y}{\dot{\gamma}} + K\,\dot{\gamma}^{n-1}$$
$$\mu'(\dot{\gamma}) = -\frac{\sigma_Y}{\dot{\gamma}^2} + (n-1)\,K\,\dot{\gamma}^{n-2}$$

- Cross Law Fluid

$$\mu(\dot{\gamma}) = \frac{\eta_0}{1 + (\lambda\dot{\gamma})^m}$$
$$\mu'(\dot{\gamma}) = -\frac{\eta_0\,\lambda\,m\,(\lambda\dot{\gamma})^{m-1}}{[1 + (\lambda\dot{\gamma})^m]^2}$$

- Arrhenius Law Fluid

$$\mu(\dot{\gamma}) = \eta\, e^{\alpha(\frac{1}{T} - \frac{1}{T_\alpha})}$$
$$\mu'(\dot{\gamma}) = 0$$

- Modified Arrhenius Law Fluid

$$\mu(\dot{\gamma}) = \eta\, e^{-\alpha(T - T_\alpha)}$$
$$\mu'(\dot{\gamma}) = 0$$

The above derivatives are easy to obtain but from a numerical point of view some of the derivatives may cause numerical difficulties as they involve singularities. These singularities can be regularised as described in chapter 3.

$\frac{d\dot{\gamma}}{d\boldsymbol{\varepsilon}}$: This term is the same for every fluid model and can be calculated by using the definition for $\dot{\gamma}$ in the following way

$$\dot{\gamma} = (2\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon})^{1/2} = \sqrt{2} \cdot \|\boldsymbol{\varepsilon}\|$$
$$\frac{d\dot{\gamma}}{d\boldsymbol{\varepsilon}} = \sqrt{2} \cdot \frac{d}{d\boldsymbol{\varepsilon}}(\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon})^{1/2} = \sqrt{2} \cdot \frac{\boldsymbol{\varepsilon}}{\|\boldsymbol{\varepsilon}\|}$$

When the above results are substituted back to the original equations the following is obtained

$$\mathcal{D}\mathbf{K}[\Delta\mathbf{u}^h] = \int_\Omega 2\alpha\mu(\mathbf{u}^h_{n+\alpha})\,\boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\Delta\mathbf{u}^h)\,\mathrm{d}\Omega + \\ + \int_\Omega 2\alpha\mu' \left[\sqrt{2}\frac{\boldsymbol{\varepsilon}}{\|\boldsymbol{\varepsilon}\|} : \boldsymbol{\varepsilon}(\Delta\mathbf{u}^h)\right] \cdot \left[\boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha})\right]\,\mathrm{d}\Omega$$

after rearranging

$$\mathcal{D}\mathbf{K}[\Delta\mathbf{u}^h] = \int_\Omega 2\alpha\mu(\mathbf{u}^h_{n+\alpha})\,\boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\Delta\mathbf{u}^h)\,\mathrm{d}\Omega + \\ + \int_\Omega \frac{4\alpha\mu'}{\dot{\gamma}} \left[\boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha}) : \boldsymbol{\varepsilon}(\Delta\mathbf{u}^h)\right]\left[\boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\varepsilon}(\mathbf{u}^h_{n+\alpha})\right]\,\mathrm{d}\Omega = \mathbf{K}^1 + \mathbf{K}^2$$

This yields two terms that yield the following after the shape functions are substituted

$$\begin{aligned}
\mathbf{K}^1 &\to \int_\Omega 2\alpha\mu\,\boldsymbol{\varepsilon}(\mathrm{N_A}\mathbf{w_A}) : \boldsymbol{\varepsilon}(\mathrm{N_B}\Delta\mathbf{u_B})\,\mathrm{d}\Omega \\
&= \int_\Omega 0.5\alpha\mu\,[\mathbf{w_A}\otimes\nabla\mathrm{N_A} + \nabla\mathrm{N_A}\otimes\mathbf{w_A}] : [\Delta\mathbf{u_B}\otimes\nabla\mathrm{N_B} + \nabla\mathrm{N_B}\otimes\Delta\mathbf{u_B}]\,\mathrm{d}\Omega \\
&= \int_\Omega 0.5\alpha\mu \Big[(\mathbf{w_A}\otimes\nabla\mathrm{N_A}) : (\Delta\mathbf{u_B}\otimes\nabla\mathrm{N_B}) + (\mathbf{w_A}\otimes\nabla\mathrm{N_A}) : (\nabla\mathrm{N_B}\otimes\Delta\mathbf{u_B}) + \\
&\quad (\nabla\mathrm{N_A}\otimes\mathbf{w_A}) : (\Delta\mathbf{u_B}\otimes\nabla\mathrm{N_B}) + (\nabla\mathrm{N_A}\otimes\mathbf{w_A}) : (\nabla\mathrm{N_B}\otimes\Delta\mathbf{u_B})\Big]\,\mathrm{d}\Omega \\
&= \int_\Omega 0.5\alpha\mu \Big[\mathbf{w_A}\cdot(\Delta\mathbf{u_B}\otimes\nabla\mathrm{N_B})\nabla\mathrm{N_A} + \mathbf{w_A}\cdot(\nabla\mathrm{N_B}\otimes\Delta\mathbf{u_B})\nabla\mathrm{N_A} + \\
&\quad \mathbf{w_A}\cdot(\nabla\mathrm{N_B}\otimes\Delta\mathbf{u_B})\nabla\mathrm{N_A} + \mathbf{w_A}\cdot(\Delta\mathbf{u_B}\otimes\nabla\mathrm{N_B})\nabla\mathrm{N_A}\Big]\,\mathrm{d}\Omega
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}^1 &\to \int_\Omega \alpha\mu \Big[\mathbf{w_A}\cdot(\Delta\mathbf{u_B}\otimes\nabla\mathrm{N_B})\nabla\mathrm{N_A} + \mathbf{w_A}\cdot(\nabla\mathrm{N_B}\otimes\Delta\mathbf{u_B})\nabla\mathrm{N_A}\Big]\,\mathrm{d}\Omega \\
&= \mathbf{w_A}\cdot\int_\Omega \alpha\mu \Big[(\nabla\mathrm{N_A}\cdot\nabla\mathrm{N_B})\Delta\mathbf{u_B} + (\nabla\mathrm{N_B}\otimes\nabla\mathrm{N_A})\Delta\mathbf{u_B}\Big]\,\mathrm{d}\Omega \\
&= \mathbf{w_A}\cdot\int_\Omega \alpha\mu \Big[(\nabla\mathrm{N_A}\cdot\nabla\mathrm{N_B})\mathbf{I} + (\nabla\mathrm{N_B}\otimes\nabla\mathrm{N_A})\Big]\,\mathrm{d}\Omega\Delta\mathbf{u_B}
\end{aligned}$$

All the values inside the integral are constant therefore the integration can be simply evaluated and yields

$$\mathbf{K}^1 \rightarrow \mathbf{w}_\mathrm{A} \cdot A\,\alpha\,\mu \left[(\nabla \mathrm{N}_\mathrm{A} \cdot \nabla \mathrm{N}_\mathrm{B})\mathbf{I} + (\nabla \mathrm{N}_\mathrm{B} \otimes \nabla \mathrm{N}_\mathrm{A}) \right] \Delta \mathbf{u}_\mathrm{B}$$

The second term is evaluated as follows

$$\begin{aligned}
\mathbf{K}^2 &\rightarrow \int_\Omega \frac{4\alpha\mu'}{\dot{\gamma}} \left[\boldsymbol{\varepsilon}(\mathbf{u}) \,:\, \boldsymbol{\varepsilon}(\mathrm{N}_\mathrm{A}\mathbf{w}_\mathrm{A}) \right] \left[\boldsymbol{\varepsilon}(\mathbf{u}) \,:\, \boldsymbol{\varepsilon}(\mathrm{N}_\mathrm{B}\Delta\mathbf{u}_\mathrm{B}) \right] \mathrm{d}\Omega \\
&= \int_\Omega \frac{4\alpha\mu'}{\dot{\gamma}} \left[\boldsymbol{\varepsilon}(\mathbf{u}) \,:\, 0.5\,(\mathbf{w}_\mathrm{A} \otimes \nabla \mathrm{N}_\mathrm{A} + \nabla \mathrm{N}_\mathrm{A} \otimes \mathbf{w}_\mathrm{A}) \right] \\
&\quad \left[\boldsymbol{\varepsilon}(\mathbf{u}) \,:\, 0.5\,(\Delta\mathbf{u}_\mathrm{B} \otimes \nabla \mathrm{N}_\mathrm{B} + \nabla \mathrm{N}_\mathrm{B} \otimes \Delta\mathbf{u}_\mathrm{B}) \right] \mathrm{d}\Omega \\
&= \int_\Omega \frac{\alpha\mu'}{\dot{\gamma}} \left[\boldsymbol{\varepsilon}(\mathbf{u}) \,:\, (\mathbf{w}_\mathrm{A} \otimes \nabla \mathrm{N}_\mathrm{A}) + \boldsymbol{\varepsilon}(\mathbf{u}) \,:\, (\nabla \mathrm{N}_\mathrm{A} \otimes \mathbf{w}_\mathrm{A}) \right] \\
&\quad \left[\boldsymbol{\varepsilon}(\mathbf{u}) \,:\, (\Delta\mathbf{u}_\mathrm{B} \otimes \nabla \mathrm{N}_\mathrm{B} + \boldsymbol{\varepsilon}(\mathbf{u}) \,:\, (\nabla \mathrm{N}_\mathrm{B} \otimes \Delta\mathbf{u}_\mathrm{B}) \right] \mathrm{d}\Omega \\
&= \int_\Omega \frac{\alpha\mu'}{\dot{\gamma}} \left[\mathbf{w}_\mathrm{A} \cdot \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{A}) + \mathbf{w}_\mathrm{A} \cdot \boldsymbol{\varepsilon}^T \nabla \mathrm{N}_\mathrm{A} \right] \\
&\quad \left[\Delta\mathbf{u}_\mathrm{B} \cdot \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{B} + \Delta\mathbf{u}_\mathrm{B} \cdot \boldsymbol{\varepsilon}^T \nabla \mathrm{N}_\mathrm{B} \right] \mathrm{d}\Omega
\end{aligned}$$

$$\begin{aligned}
\mathbf{K}^2 &\rightarrow \int_\Omega \frac{4\alpha\mu'}{\dot{\gamma}} (\mathbf{w}_\mathrm{A} \cdot \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{A})(\Delta\mathbf{u}_\mathrm{B} \cdot \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{B})\, \mathrm{d}\Omega \\
&= \mathbf{w}_\mathrm{A} \cdot \int_\Omega \frac{4\alpha\mu'}{\dot{\gamma}} (\Delta\mathbf{u}_\mathrm{B} \cdot \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{B}) \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{A}\, \mathrm{d}\Omega \\
&= \mathbf{w}_\mathrm{A} \cdot \int_\Omega \frac{4\alpha\mu'}{\dot{\gamma}} (\boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{A} \otimes \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{B})\, \mathrm{d}\Omega \Delta\mathbf{u}_\mathrm{B}
\end{aligned}$$

A simple integration rule can be applied to this term as well, leading to

$$\mathbf{K}^2 \rightarrow \mathbf{w}_\mathrm{A} \cdot \frac{4\,A\,\alpha\,\mu'}{\dot{\gamma}} (\boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{A} \otimes \boldsymbol{\varepsilon} \nabla \mathrm{N}_\mathrm{B}) \Delta\mathbf{u}_\mathrm{B}$$

5.5.6 Derivation of the Stiffness matrix - Part $\mathbf{G}_p$

This term represents the pressure part of the stress tensor, and reads

$$\mathbf{G}_p = -\int_\Omega p^h_{n+\alpha}\, \boldsymbol{\varepsilon}(\mathbf{w}^h) \,:\, \mathbf{I}\, \mathrm{d}\Omega$$

The equations are solved for the variables at time $n+1$ which means that $p^h_{n+\alpha}$ is a function of p^h_{n+1} The directional derivative of $p^h_{n+\alpha}$ in the direction Δp^h is obtained as

$$\mathcal{D}p^h_{n+\alpha}[\Delta p^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} p^h_{n+\alpha}(p^h_{n+1}) = \alpha\Delta p^h \quad \Rightarrow \quad \mathcal{D}p^h_{n+\alpha}[\Delta p_{\mathrm{B}}] = \alpha \mathrm{N_B}\Delta p_{\mathrm{B}} \tag{5.40}$$

The derivative of $\mathbf{G}_p$ can then be evaluated as follows

$$\begin{aligned}
\mathcal{D}\mathbf{G}_p[\Delta p_{\mathrm{B}}] \rightarrow & -\int_\Omega \alpha\Delta p_{\mathrm{B}}\mathrm{N_B}0.5\ (\mathbf{w}_{\mathrm{A}} \otimes \nabla \mathrm{N_A} + \nabla \mathrm{N_A} \otimes \mathbf{w}_{\mathrm{A}}) : \mathbf{I}\, \mathrm{d}\Omega \\
&= -\int_\Omega \alpha\Delta p_{\mathrm{B}}\ \mathrm{N_B}\ 0.5 \times 2\ \mathbf{w}_{\mathrm{A}} \cdot \nabla \mathrm{N_A}\, \mathrm{d}\Omega \\
&= \mathbf{w}_{\mathrm{A}} \cdot \alpha \left[-\int_\Omega \mathrm{N_B}\, \mathrm{d}\Omega\ \nabla \mathrm{N_A} \right] \Delta p_{\mathrm{B}} \qquad (\nabla \mathrm{N_A} \text{ is constant})
\end{aligned}$$

By using (5.10) the above can be integrated yielding

$$\mathcal{D}\mathbf{G}_p[\Delta p_{\mathrm{B}}] = \mathbf{w}_{\mathrm{A}} \cdot \alpha \left[-\frac{Ad}{3} \nabla \mathrm{N_A} \right] \Delta p_{\mathrm{B}}$$

5.5.7 Derivation of the Stiffness matrix - Part $\mathbf{G}^{\mathrm{T}}$

This term represents the incompressibility condition

$$\mathbf{G}^{\mathrm{T}} = \int_\Omega \mathrm{q}^h\ \nabla \cdot \mathbf{u}^h_{n+\alpha}\, \mathrm{d}\Omega$$

The term can be integrated by using the same methodology as earlier resulting in

$$\begin{aligned}
\mathcal{D}\mathbf{G}^{\mathrm{T}}[\Delta \mathbf{u}_{\mathrm{B}}] \rightarrow & \ \mathrm{q_A} \int_\Omega \mathrm{N_A} \nabla \cdot (\mathrm{N_B}\ \Delta \mathbf{u}_{\mathrm{B}})\, \mathrm{d}\Omega \\
&= \mathrm{q_A} \int_\Omega \mathrm{N_A} \nabla \mathrm{N_B} \cdot \Delta \mathbf{u}_{\mathrm{B}}\, \mathrm{d}\Omega \\
&= \mathrm{q_A} \int_\Omega \mathrm{N_A}\, \mathrm{d}\Omega\ \nabla \mathrm{N_B} \cdot \Delta \mathbf{u}_{\mathrm{B}} \qquad (\nabla \mathrm{N_B} \text{ is constant}) \\
&= \mathrm{q_A} \left[\frac{Ad}{3} \nabla \mathrm{N_B} \right] \cdot \Delta \mathbf{u}_{\mathrm{B}}
\end{aligned}$$

5.5.8 Derivation of the Stiffness matrix - Part $\mathbf{G}_\delta$

This term represents the streamline upwind contribution to the pressure part of the stress in the governing equations, and is given by

$$\mathbf{G}_\delta = \int_{\Omega_e} \tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma} \cdot \nabla)\mathbf{w}^h \cdot (\nabla \cdot p^h_{n+\alpha}\,\mathbf{I})\,\mathrm{d}\Omega$$

The term involves three factors that are dependent on the unknown variables. After applying the product rule this will lead to three parts the stiffness matrix. Thus

$$\begin{aligned}
\mathcal{D}\mathbf{G}_\delta[\Delta\mathbf{u}^h, \Delta p^h] &= \left.\frac{d}{d\zeta}\right|_{\zeta=0} \int_{\Omega_e} \tau(\mathbf{u}^h_{n+\gamma})(\mathbf{u}^h_{n+\gamma} \cdot \nabla)\mathbf{w}^h\,[\nabla \cdot p^h_{n+\alpha}\,\mathbf{I}]\,\mathrm{d}\Omega \\
&= \int_{\Omega_e} (\boldsymbol{\tau}' \cdot \Delta\mathbf{u}^h)(\mathbf{u}^h_{n+\gamma} \cdot \nabla)\mathbf{w}^h \cdot (\nabla \cdot p^h_{n+\alpha}\,\mathbf{I}) \\
&\quad + \gamma\tau(\Delta\mathbf{u}^h \cdot \nabla)\mathbf{w}^h \cdot (\nabla \cdot p^h_{n+\alpha}\,\mathbf{I}) \\
&\quad + \alpha\tau(\mathbf{u}^h_{n+\gamma} \cdot \nabla)\mathbf{w}^h \cdot (\nabla \cdot \Delta p^h\,\mathbf{I})\,\mathrm{d}\Omega
\end{aligned}$$

As with other term further simplification will be done after introducing the shape functions, i.e.

$$\begin{aligned}
\mathcal{D}\mathbf{G}_\delta[\Delta\mathbf{u}_\mathrm{B}, \Delta p_\mathrm{B}] \rightarrow &\int_{\Omega_e} (\boldsymbol{\tau}'_\mathrm{B} \cdot \Delta\mathbf{u}_\mathrm{B})(\mathbf{u}_{n+\gamma} \cdot \nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A} \cdot (\nabla \cdot p_{n+\alpha}\,\mathbf{I}) \\
&+ \gamma\tau(\mathrm{N_B}\Delta\mathbf{u}_\mathrm{B} \cdot \nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A} \cdot (\nabla \cdot p_{n+\alpha}\,\mathbf{I}) \\
&+ \alpha\tau(\mathbf{u}_{n+\gamma} \cdot \nabla)\mathrm{N_A}\mathbf{w}_\mathrm{A} \cdot (\nabla \cdot \mathrm{N_B}\Delta p_\mathrm{B}\,\mathbf{I})\,\mathrm{d}\Omega \\
= &\int_{\Omega_e} (\boldsymbol{\tau}'_\mathrm{B} \cdot \Delta\mathbf{u}_\mathrm{B})(\mathbf{u}_{n+\gamma} \cdot \nabla\mathrm{N_A})\mathbf{w}_\mathrm{A} \cdot (\nabla \cdot p_{n+\alpha}\,\mathbf{I}) \\
&+ \gamma\tau\mathrm{N_B}(\Delta\mathbf{u}_\mathrm{B} \cdot \nabla\mathrm{N_A})\mathbf{w}_\mathrm{A} \cdot (\nabla \cdot p_{n+\alpha}\,\mathbf{I}) \\
&+ \alpha\tau(\mathbf{u}_{n+\gamma} \cdot \nabla\mathrm{N_A})\mathbf{w}_\mathrm{A} \cdot \nabla\mathrm{N_B}\Delta p_\mathrm{B}\,\mathrm{d}\Omega
\end{aligned}$$

The transformation of the above form is now performed in several steps with the main purpose of separating the stiffness matrix from the increments $\Delta\mathbf{u}_\mathrm{B}$ and Δp_B, as follows

$$
\begin{aligned}
\mathcal{D}\mathbf{G}_\delta[\Delta\mathbf{u}_\mathrm{B}, \Delta p_\mathrm{B}] = \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} & (\mathbf{u}_{n+\gamma} \cdot \nabla \mathrm{N}_\mathrm{A})(\boldsymbol{\tau}'_\mathrm{B} \cdot \Delta\mathbf{u}_\mathrm{B}) \nabla p_{n+\alpha} \\
& + \gamma\tau \mathrm{N}_\mathrm{B}(\Delta\mathbf{u}_\mathrm{B} \cdot \nabla \mathrm{N}_\mathrm{A}) \nabla p_{n+\alpha} \\
& + \alpha\tau(\mathbf{u}_{n+\gamma} \cdot \nabla \mathrm{N}_\mathrm{A}) \nabla \mathrm{N}_\mathrm{B} \Delta p_\mathrm{B} \, \mathrm{d}\Omega \\
= \quad & \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} (\mathbf{u}_{n+\gamma} \cdot \nabla \mathrm{N}_\mathrm{A})(\nabla p_{n+\alpha} \otimes \boldsymbol{\tau}'_\mathrm{B}) \, \mathrm{d}\Omega \Delta\mathbf{u}_\mathrm{B} \\
& + \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} \gamma\tau \mathrm{N}_\mathrm{B}(\nabla p_{n+\alpha} \otimes \nabla \mathrm{N}_\mathrm{A}) \, \mathrm{d}\Omega \Delta\mathbf{u}_\mathrm{B} \\
& + \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} \alpha\tau(\mathbf{u}_{n+\gamma} \cdot \nabla \mathrm{N}_\mathrm{A}) \nabla \mathrm{N}_\mathrm{B} \, \mathrm{d}\Omega \Delta p_\mathrm{B} \\
= \quad & \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} \tau(\mathbf{u}_{n+\gamma} \cdot \nabla \mathrm{N}_\mathrm{A}) \left[\nabla p_{n+\alpha} \otimes \frac{\boldsymbol{\tau}'_\mathrm{B}}{\tau} \right] \mathrm{d}\Omega \Delta\mathbf{u}_\mathrm{B} \\
& + \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} \gamma\tau \mathrm{N}_\mathrm{B}(\nabla p_{n+\alpha} \otimes \nabla \mathrm{N}_\mathrm{A}) \, \mathrm{d}\Omega \Delta\mathbf{u}_\mathrm{B} \\
& + \mathbf{w}_\mathrm{A} \cdot \int_{\Omega_e} \alpha\tau(\mathbf{u}_{n+\gamma} \cdot \nabla \mathrm{N}_\mathrm{A}) \nabla \mathrm{N}_\mathrm{B} \, \mathrm{d}\Omega \Delta p_\mathrm{B}
\end{aligned}
$$

The integration of the expression is illustrated for the 2D situation and is performed by employing three gauss points. The variables that change their values at gauss points are denoted by a subscript *gp*. Hence

$$
\begin{aligned}
\mathcal{D}\mathbf{G}_\delta[\Delta\mathbf{u}_\mathrm{B}, \Delta p_\mathrm{B}] = \quad & \mathbf{w}_\mathrm{A} \cdot \frac{Ad}{3} \sum_{gp=1}^{NGauss} \tau(\mathbf{u}_{n+\gamma}^{gp} \cdot \nabla \mathrm{N}_\mathrm{A}) \left[\nabla p_{n+\alpha} \otimes \frac{\boldsymbol{\tau}_\mathrm{B}^{\prime gp}}{\tau} \right] \Delta\mathbf{u}_\mathrm{B} \\
& + \mathbf{w}_\mathrm{A} \cdot \frac{Ad}{3} \sum_{gp=1}^{NGauss} \gamma\tau \mathrm{N}_\mathrm{B}^{gp}(\nabla p_{n+\alpha} \otimes \nabla \mathrm{N}_\mathrm{A}) \Delta\mathbf{u}_\mathrm{B} \\
& + \mathbf{w}_\mathrm{A} \cdot \frac{Ad}{3} \sum_{gp=1}^{NGauss} \alpha\tau(\mathbf{u}_{n+\gamma}^{gp} \cdot \nabla \mathrm{N}_\mathrm{A}) \nabla \mathrm{N}_\mathrm{B} \Delta p_\mathrm{B} \\
= \quad & \mathbf{G}_\delta^1 + \mathbf{G}_\delta^2 + \mathbf{G}_\delta^3
\end{aligned}
$$

5.5.9 Derivation of the Stiffness matrix - Part $\mathbf{G}_\epsilon$

The last part of the stiffness matrix involves the pressure stabilisation of the pressure term, and is given as

$$
\mathbf{G}_\epsilon = \int_{\Omega_e} \tau \frac{1}{\rho} \nabla \mathrm{q}^h \cdot (\nabla \cdot p_{n+\alpha}^h \, \mathbf{I}) \, \mathrm{d}\Omega
$$

The directional derivative is performed with respect to both velocity and pressure variables and reads

$$\mathcal{D}\mathbf{G}_\epsilon[\Delta\mathbf{u}^h, \Delta p^h] = \frac{1}{\rho}\frac{d}{d\zeta}\bigg|_{\zeta=0} \int_{\Omega_e} \tau(\mathbf{u}^h_{n+\gamma})\nabla \mathrm{q}^h \cdot (\nabla \cdot p^h_{n+\alpha}\,\mathbf{I})\,\mathrm{d}\Omega$$
$$= \frac{1}{\rho}\int_{\Omega_e} (\boldsymbol{\tau}'_{\mathrm{B}} \cdot \Delta\mathbf{u}^h)\nabla \mathrm{q}^h \cdot (\nabla \cdot p^h_{n+\alpha}\,\mathbf{I}) + \alpha\tau\nabla \mathrm{q}^h \cdot (\nabla \cdot \Delta p^h\,\mathbf{I})\,\mathrm{d}\Omega$$

The stiffness matrix has two parts the first is a result of dependency of τ on the velocity, while the second is a derivative of the pressure. It can be written as

$$\mathcal{D}\mathbf{G}_\epsilon[\Delta\mathbf{u}_{\mathrm{B}}, \Delta p_{\mathrm{B}}] \rightarrow \frac{1}{\rho}\int_{\Omega_e} (\boldsymbol{\tau}'_{\mathrm{B}} \cdot \Delta\mathbf{u}_{\mathrm{B}})\nabla(\mathrm{N_A q_A}) \cdot (\nabla \cdot p_{n+\alpha}\,\mathbf{I})$$
$$+ \alpha\tau\nabla(\mathrm{N_A q_A}) \cdot (\nabla \cdot \mathrm{N_B}\Delta p_{\mathrm{B}}\,\mathbf{I})\,\mathrm{d}\Omega$$

The following expression is further rearranged in several steps with the aim to separate the stiffness matrix from the increments of the test parameter $\mathrm{q_A}$, the pressure Δp_{B} and velocity $\Delta\mathbf{u}_{\mathrm{B}}$. Hence

$$\mathcal{D}\mathbf{G}_\epsilon[\Delta\mathbf{u}_{\mathrm{B}}, \Delta p_{\mathrm{B}}] = \frac{1}{\rho}\int_{\Omega_e} (\boldsymbol{\tau}'_{\mathrm{B}} \cdot \Delta\mathbf{u}_{\mathrm{B}})\nabla \mathrm{N_A q_A} \cdot \nabla p_{n+\alpha}$$
$$+ \alpha\tau\nabla \mathrm{N_A q_A} \cdot \nabla \mathrm{N_B}\Delta p_{\mathrm{B}}\,\mathrm{d}\Omega$$

$$\mathcal{D}\mathbf{G}_\epsilon[\Delta\mathbf{u}_{\mathrm{B}}, \Delta p_{\mathrm{B}}] = \frac{\mathrm{q_A}}{\rho}\int_{\Omega_e} (\boldsymbol{\tau}'_{\mathrm{B}} \cdot \Delta\mathbf{u}_{\mathrm{B}})(\nabla \mathrm{N_A} \cdot \nabla p_{n+\alpha})$$
$$+ \alpha\tau(\nabla \mathrm{N_A} \cdot \nabla \mathrm{N_B})\Delta p_{\mathrm{B}}\,\mathrm{d}\Omega$$

$$\mathcal{D}\mathbf{G}_\epsilon[\Delta\mathbf{u}_{\mathrm{B}}, \Delta p_{\mathrm{B}}] = \frac{\mathrm{q_A}}{\rho}\int_{\Omega_e} (\nabla \mathrm{N_A} \cdot \nabla p_{n+\alpha})\boldsymbol{\tau}'_{\mathrm{B}}\,\mathrm{d}\Omega \cdot \Delta\mathbf{u}_{\mathrm{B}}$$
$$+ \frac{\mathrm{q_A}}{\rho}\int_{\Omega_e} \alpha\tau(\nabla \mathrm{N_A} \cdot \nabla \mathrm{N_B})\,\mathrm{d}\Omega\Delta p_{\mathrm{B}}$$

The integration step can then be executed as follows

$$\mathcal{D}\mathbf{G}_\epsilon[\Delta\mathbf{u}^h, \Delta p_{\mathrm{B}}] = \mathrm{q_A}\left[\frac{A\tau d}{3\rho}(\nabla \mathrm{N_A} \cdot \nabla p)\sum_{gp=1}^{NGauss} \frac{\boldsymbol{\tau}'^{gp}_{\mathrm{B}}}{\tau}\right] \cdot \Delta\mathbf{u}_{\mathrm{B}} +$$
$$\mathrm{q_A}[\frac{\alpha A d}{\rho}(\nabla \mathrm{N_A} \cdot \nabla \mathrm{N_B})\tau]\Delta p_{\mathrm{B}}$$
$$= \mathbf{G}^1_\epsilon + \mathbf{G}^2_\epsilon$$

The first term is integrated by using the Gauss integration rule which is a consequence of $\boldsymbol{\tau}'_{\mathrm{B}}$ not being a constant. All the factors in the second part are constant which means it can be integrated directly.

5.6 Implementation

In the previous sections the formulae for internal forces and stiffness matrix have been determined. It is now necessary to implement them into a finite element program. The element routines where implemented in the finite element program "ELFEN". Four different elements where implemented. These include steady state and transient elements for both 2D and 3D non-Newtonian fluid flow. 2D elements are also incorporated with options for axisymmetric flows. The axisymmetric flows can be axisymmetric either about the x axis or the y axis, which is achieved by changing the thickness appropriately. Details can by found in section 5.2.1.

Different non-Newtonian fluid models are implemented in a separate routine that is used by all the element routines. The various terms that make up the internal forces and the stiffness matrix can be divided into two groups. The first group involves terms that have been integrated explicitly and do not require numerical integration. The second group involves terms that need to be integrated numerically. The two groups are calculated in two separate loops over nodes. The first group is also symmetric. This enables us to evaluate only half of the stiffness matrix for this group. Details of the implementation can be found in Appendix D.

There is a large number of terms that are involved in this implementation. This introduces additional difficulties both in terms of the volume of manual work that needs to be performed and the increasing possibility of an error occurring. This has been overcome by a script that automates this process. The scripting language is part of the "Mathematica" package. Details are provided in Appendix C.

A series of tests have been performed to verify the described strategy for modelling of non-Newtonian fluid flows. These tests are elaborated in the next chapter. They are ordered by complexity and finish with large scale problems.

Chapter 6

Numerical Examples

To verify the finite element discretization several tests were performed. In order to verify the Bingham and Power law model a Couette flow test is used in which the numerical solutions are compared to analytical solutions. Secondly an extrusion problem is performed and the numerical solution is compared to the classical plastic flow solution. Next 2D and 3D test problems that describe flow around a cylinder and a sphere respectively are performed. Finally large scale 3D problems associated with the industrial practice are reported.

6.1 Couette flow test

A Couette flow is a channel flow that is often coupled with a shearing flow. Couette flow is one of the rare problems for which an analytical solution can be found in a closed form. This makes it ideal for testing various aspects of fluid modelling. The analytical solutions for a power law and Bingham fluids have been described in Chapter 3.6.

6.1.1 Bingham fluid

The analytical solution for the Bingham fluid is obtained by determining a number of constants. These constants are η_1, η_2, λ_1, λ_2, u_{max} and C_2^3. The procedure for obtaining them for a given value of the pressure gradient and the wall velocity is outlined in section 3.6.2. Once these constants are found velocity function (3.23) can be used to get analytical values for the velocity distribution across the channel section. These values can be then compared to numerical ones.

To obtain the numerical values for Couette flow the mesh in Figure 6.1 has been used. This mesh consists of 15484 elements and 8015 nodes. The mesh has been refined along the cross-section in the centre of mesh. The reason for this refinement is to provide fine resolution for the velocity values along this line. The horizontal size of the domain has been chosen to insure that the effect of channel

boundaries is minimised. The fluid properties that were used can be found in Table 6.1. Convergence for a number of pressure gradients can be found in Table 6.2.

Fluid properties

Fluid model name	Bingham plastic
Plastic viscosity $\mu[Pa \cdot s]$	10.0
Yield Stress $\sigma_Y[Pa]$	100.0
Density $\rho[kg/m^3]$	1.0
Numerical properties	
Stress growth exponent[1] $m[-]$	300.0

[1] see JNNFM (Blackery & Mitsoulis, 1997)

Table 6.1: Constitutive properties for Bingham fluid

Euclidean norm

ITERATION	$t = 0.05s$ $\Delta p = 1000Pa$	$t = 0.25s$ $\Delta p = 5000Pa$	$t = 0.30s$ $\Delta p = 5000Pa$	$t = 1.20s$ $\Delta p = 5000Pa$
1	0.831390E-01	0.160941E-01	0.136376E-04	0.168634E-04
2	0.346958E-04	0.103413E-03	0.554616E-10	0.151522E-09
3	0.134515E-10	0.488251E-08		

Table 6.2: Convergence table

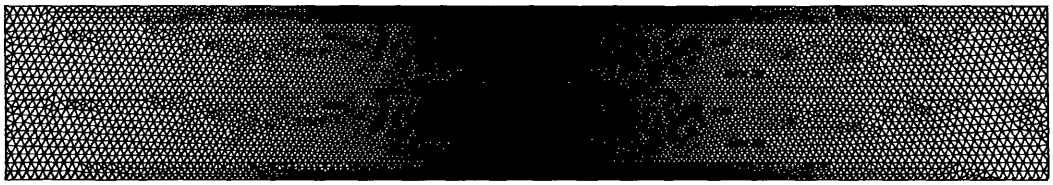


Figure 6.1: Finite element mesh for the Couette example

A comparison between the analytical values and numerical values for a number of different pressure gradients can be found in Figure 6.2.

6.1.2 Power law fluid

The Couette flow of a power law fluid represents the second test that was used to verify the numerical model. The mesh shown in Figure 6.1 is used. The fluid properties for this model can be found in Table 6.3. The comparison between analytical values and numerical values can be found in Figure 6.3. It can be observed that the numerical solution almost exactly reproduces the analytical solution.

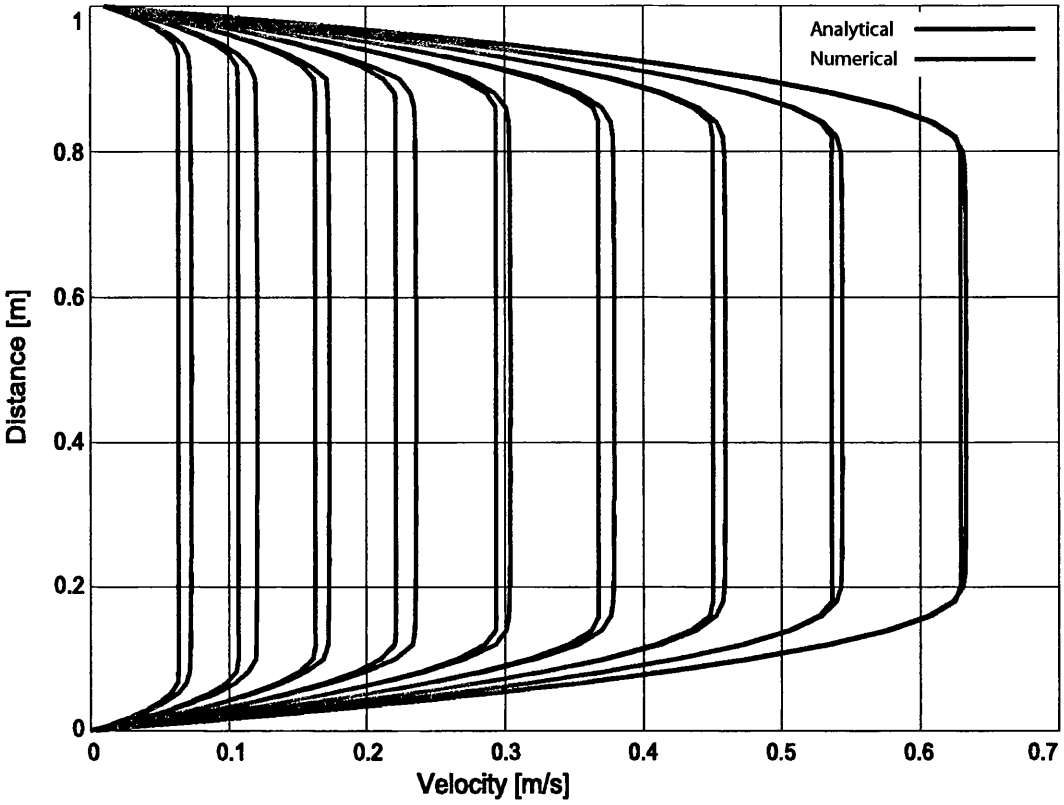


Figure 6.2: Comparison between numerical and analytical values for velocity distribution of a Bingham fluid

Fluid properties

Fluid model name	Power law
Fluid consistency index $m[Pa \cdot s^n]$	0.84
Power law index $n[-]$	0.5088
Density $\rho[kg/m^3]$	1.0
Numerical properties	
Critical shear-rate $\dot{\gamma}_0[s^{-1}]$	1E-5

Table 6.3: Constitutive properties for power law fluid

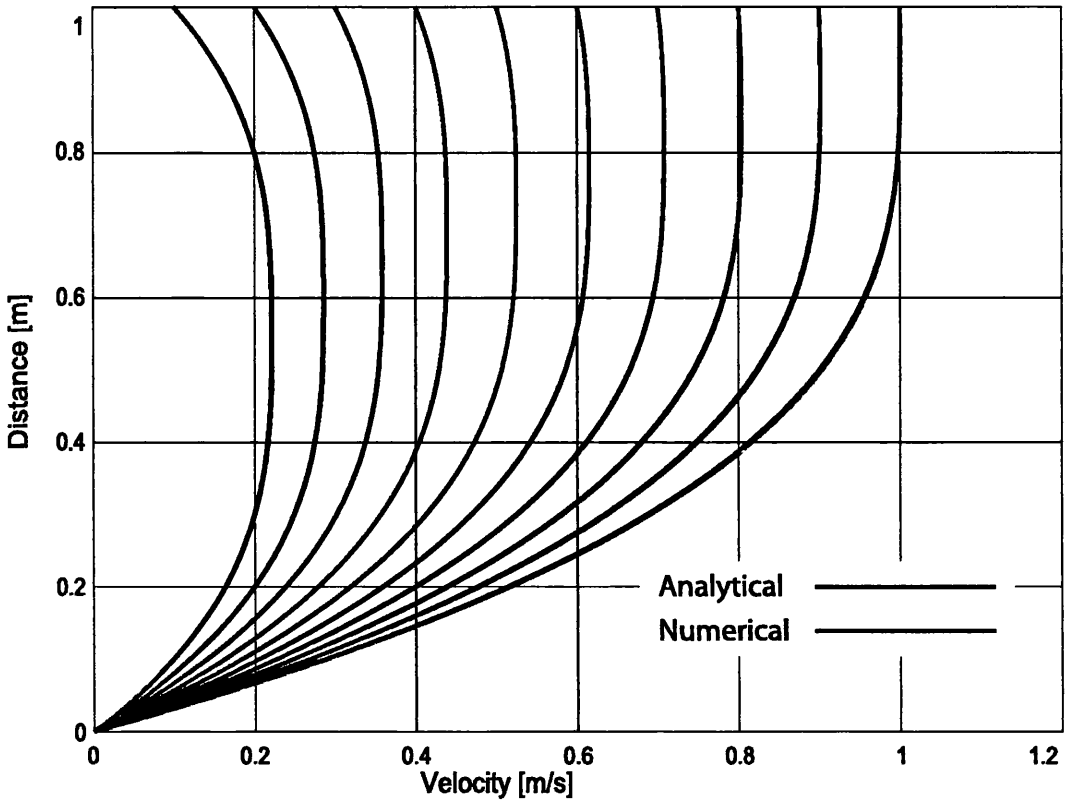


Figure 6.3: Comparison between numerical and analytical values for velocity distribution of a power law fluid

6.2 A 2D ram extrusion

This is the first of several numerical tests that explore the various aspects of extrusion. Before the details of the first test are presented an overview of extrusion as an industrial process will be given.

6.2.1 Extrusion

Extrusion is one of the basic shaping processes used to shape metals, ceramics and polymers. It is a compression process in which the material is forced through a die. The purpose of extrusion is to provide long continuous products whose cross-sectional area is constant and determined by the die orifice. The extrusion process is widely used for thermoplastics and elastomers to produce pipes, tubes, windows, door frames and coated electrical wires. For metals it is used to produce wire and various steel and aluminium cross sections. Production is organised as a continuous process in which the extruded product is cut to length in subsequent operations.

When the extrusion process is used to extrude polymers the extrudate is fed to the die by means of screw extruder. The screw extruder consists of a screw and barrel. The barrel of the extruder has on one end a hopper that is fed with polymer pellets. These pellets are then transported via a drag flow mechanism though several stages before the polymer is converted into a polymer melt and is finally fed through the die. The stages that the polymer goes through are heating, metering and compression. The heating stage is used to melt the polymer pellets. After the polymer has been melted the metering stage is employed to homogenise the melt. This is achieved by a kneading action. Finally the compression stage provides sufficient pressure for the die. Between the extruder and the die a breaker plate may be inserted that straightens the flow and stops hard lumps from entering the die. The breaker plate consists of small axial holes. Extruders can either have a single screw or have twin intermeshing screws. Twin screw extruders are used if greater mixing is required.

The extrusion of metals is somewhat different than polymer extrusion. The metal is formed as a billet that is then forced through the die by using a ram. The billet is usually of a cylindrical form. Dies used in the extrusion process differ according to the type of metal that is extruded. In the case of aluminium extrusion the die is flat faced with dead metal zones near the die entry. A zone of shearing develops between the dead zones and the moving extrudate as the extrusion progresses. The extrusion process can be direct or indirect. In the case of direct extrusion the billet is forced through a stationary die where as for an indirect extrusion the die is immersed into the stationary billet. For the extrusion of steel alloys of titanium and nickel a shaped die is used. The die is shaped in a manner that insures gradual reduction of the cross sectional area. For this type of extrusion a lubricant has to be used. This is not necessary for extrusions that involve aluminium.

One thing that is common for all extrusion processes is that the equipment

used varies a lot depending on the materials used and the desired end product. This means that the process needs to be constantly updated and optimised with respect to production needs. The other characteristic is that extrusion processes exhibit a wide range of different physical phenomena within complex geometries. The range and complexity makes the simulation of extrusion processes challenging task. For these reason several numerical tests that involve extrusion processes were performed. Further details about extrusion can be found in Groover (1996); O'Brien (1992); Michaeli (1992); Tucker (1989); Gunasekera (1989).

6.2.2 Ram extrusion simulation

A ram extrusion is a process in which material is forced through a die using a ram to produce a reduced cross-sectional area. Ram extrusion is frequently used in manufacturing and process industries.

For the 2D geometry represented in Figure 6.4 and for a rigid plastic material analytical solutions can be obtained (Lubliner, 1990). This analytical solution yields an upper bound ram pressure. It is obtained by using slip-line theory. The upper bound for the ram force is the appropriate value to be sought because it ensures steady plastic flow. This value on the other hand can be then compared to a value obtained by numerical simulation of a Bingham fluid. The parameters for the Bingham fluid law are chosen so that the plastic viscosity is very low compared to the yield stress, (see Table 6.5a). Having a very low value for the plastic viscosity in effect makes the rate dependent part of the fluid model negligible.

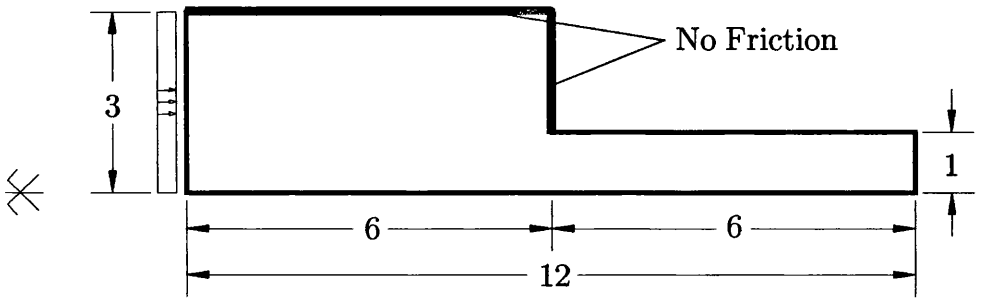


Figure 6.4: Geometry for the 2D extrusion example

The 2D ram extrusion test involves a two-thirds reduction in the cross-sectional area. Because the problem is symmetric only half of the problem is considered. The container walls in the extrusion problem are assumed to be frictionless. The dimensions, boundary conditions and loading of the problem can be found in Figure 6.4. The properties of the fluid that are used in the problem are shown in Figure 6.5(a). The pressure applied by the ram is modelled using a traction boundary condition, with the time function given in Figure 6.5(b).

a.)

Fluid model name	Bingham plastic
Plastic viscosity $\mu[Pa \cdot s]$	0.000001
Yield Stress $\sigma_Y[Pa]$	1000.0
Density $\rho[kg/m^3]$	100.0
Numerical properties	
Stress growth exponent[2] $m[-]$	1000.0

see JNNFM (Blackery & Mitsoulis, 1997)

b.)

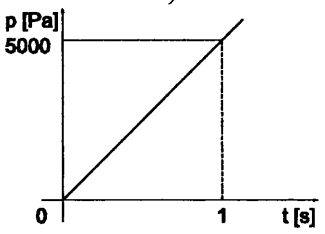


Figure 6.5: a.) Table of fluid properties b.) Ram load function

The object of the test is to obtain a slip line field and the associated value of the ram pressure. The slip lines can be observed when contour plots for effective strain rates are plotted. The time instant when the slip line occurs is used to calculate the extrusion pressure. The slip line field consists of a centred fan that extends to the corner. A diagram of the slip line field is given in Figure 6.6.

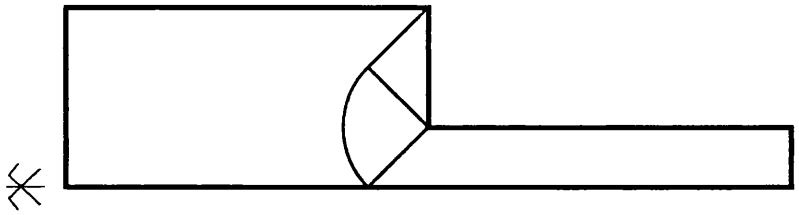


Figure 6.6: Slip line field

The numerical value of the pressure that produces the slip lines is obtained by using a ramp load function. The ramp load function has the value of $0.0Pa$ at time $0.0s$ and the value of $5000.0Pa$ at time $1.0s$, (see Table 6.5b). By observing the time instant when the slip line field occurs a value of the external pressure can be obtained. The value can also be read from Figure 6.13. This pressure can be compared to an analytical value that is computed using the classical plasticity theory.

The value of the extrusion pressure can be calculated by using the following formula (Lubliner, 1990):

$$p = \frac{4}{3}\left[1 + \frac{\pi}{2}\right] k \tag{6.1}$$

where k is the yield stress and p is the extrusion pressure.

To perform the simulation the finite element mesh shown in Figure 6.7 has been employed.

The finite element mesh consists of 10835 elements and 5586 nodes. The mesh has been refined around point "B" because of the expected complex flow field around this point. To perform the simulation 29 increments were used. To be

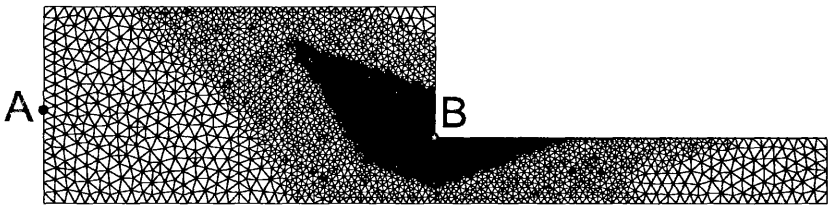


Figure 6.7: Mesh for the 2D extrusion example

able to evaluate the accuracy of the numerical simulation several coarser mesh were utilised. The coarser meshes can be seen in Figures 6.8 – 6.12.

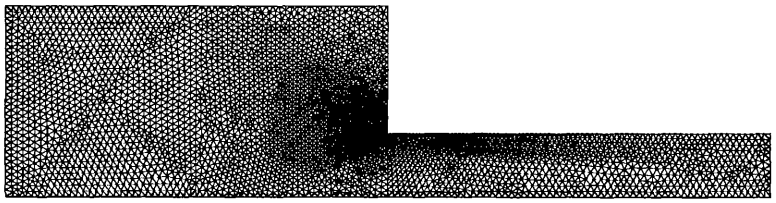


Figure 6.8: Mesh 2 for the 2D extrusion example

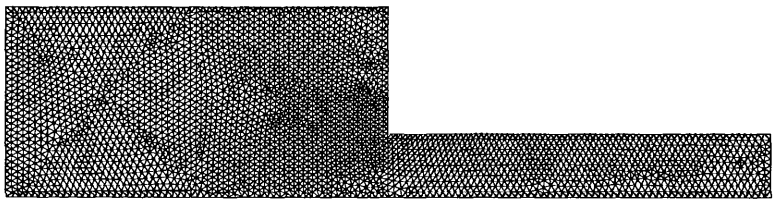


Figure 6.9: Mesh 3 for the 2D extrusion example

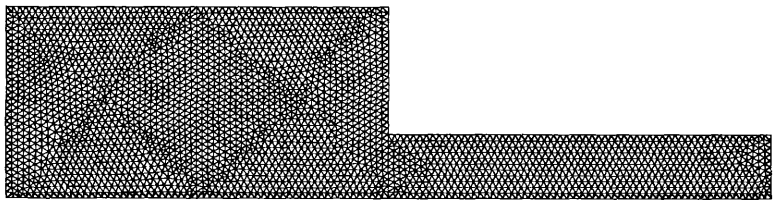


Figure 6.10: Mesh 4 for the 2D extrusion example

The convergence for some representative time steps can be observed in Table 6.4. As with the previous numerical test quadratic convergence can be observed at all stages of the test.

The occurrence of the slip line was observed at pseudo time 0.688 which in turn gives an extrusion pressure of $p_{numerical} = 0.688 \times 5000 = 3440N$. This

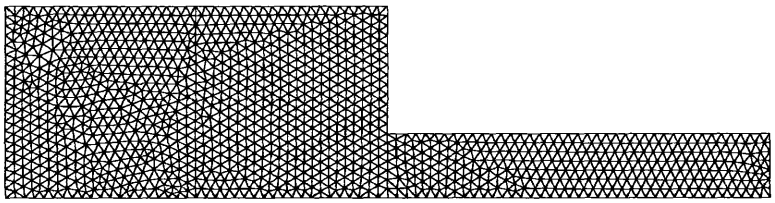


Figure 6.11: Mesh 5 for the 2D extrusion example

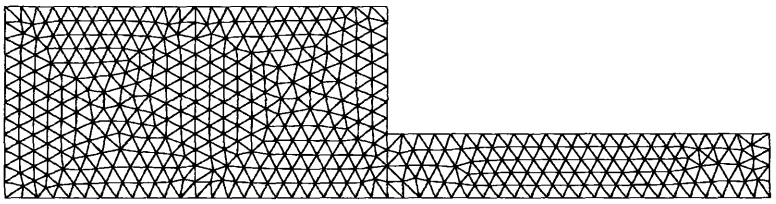


Figure 6.12: Mesh 6 for the 2D extrusion example

Euclidean norm

ITERATION	$t = 0.680s$ $\Delta t = 0.040s$	$t = 0.688s$ $\Delta t = 0.001s$	$t = 1.600s$ $\Delta t = 0.050s$	$t = 2.500s$ $\Delta t = 0.050s$
1	0.150038	0.460167	0.660952	1.57078
2	0.717860E-01	0.579845	1.03258	1.19625
3	0.215295E-01	0.301548	0.371735	0.982628
4	0.187914E-02	0.988493E-01	0.247051	1.25367
5	0.115672E-04	0.827178E-01	0.171162	0.668515
6	0.327777E-09	0.271869E-01	0.986868E-01	0.874405
7		0.638231E-02	0.210569E-01	0.513817
8		0.809384E-03	0.151909E-02	0.413222
9		0.177206E-04	0.258966E-04	0.475986
10		0.916657E-08	0.472538E-08	0.406904
11				0.261484
12				0.685478E-01
13				0.246030E-02
14				0.116935E-04
15				0.678459E-09

Table 6.4: Convergence table

pressure can be compared with the analytical value, obtained using (6.1), as

$$p_{analytical} = \frac{4}{3}\left[1 + \frac{\pi}{2}\right] 1000.0 = 3428N \tag{6.2}$$

This gives an error of only 0.35%. A comparison of extrusion pressures and relevant

errors for the various mashes can be observed in Table 6.5. It can be observed that the accuracy increases with mesh refinement and approaches the analytical solution.

Mesh	Number of elements	Number of nodes	Time instant (s)	Extrusion pressure (N)	Error (%)
1	10835	5586	0.688	3440	0.35
2	7222	3778	0.702	3510	2.39
3	6166	3240	0.708	3540	3.27
4	5358	2830	0.730	3650	6.48
5	2392	1297	0.740	3700	7.93
6	880	501	0.758	3790	10.56

Table 6.5: Comparison table

The evolution of the slip line can be observed in Figure 6.14. The pictures show two time instances before the material starts to flow whereas the remaining pictures show effective strain rate contour plots after the extrusion pressure has been reached. The third time instant is used to calculate the extrusion pressure. The first two time instances indicate the beginning of slip lines development near the tip of the die. The other contour plots clearly exhibit behaviour that is similar to the expected slip lines in Figure 6.6. Another indicator of the difference in behaviour between time instances 0.687 and 0.688 can observed by the difference in the velocity fields which is of the order of several magnitudes. A good insight into the behaviour of the material can be obtained by observing a pressure velocity graph given in Figure 6.13, which represents the time evolution of pressure and velocity at point "A" in Figure 6.7. In effect the material does not move until the extrusion pressure has been reached. Once the extrusion pressure has been reached a small change in pressure induces a large change in velocity, hence the material has started to flow.

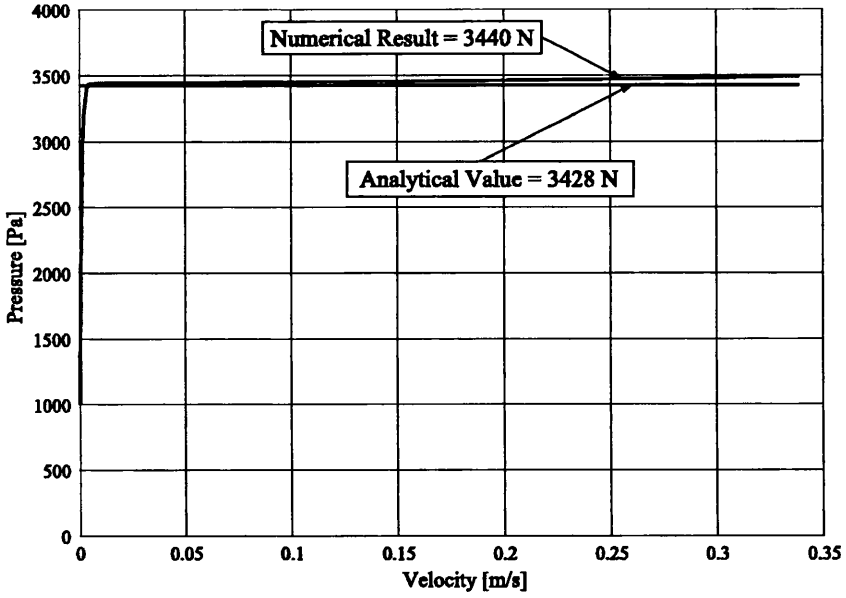


Figure 6.13: Pressure velocity graph at point A

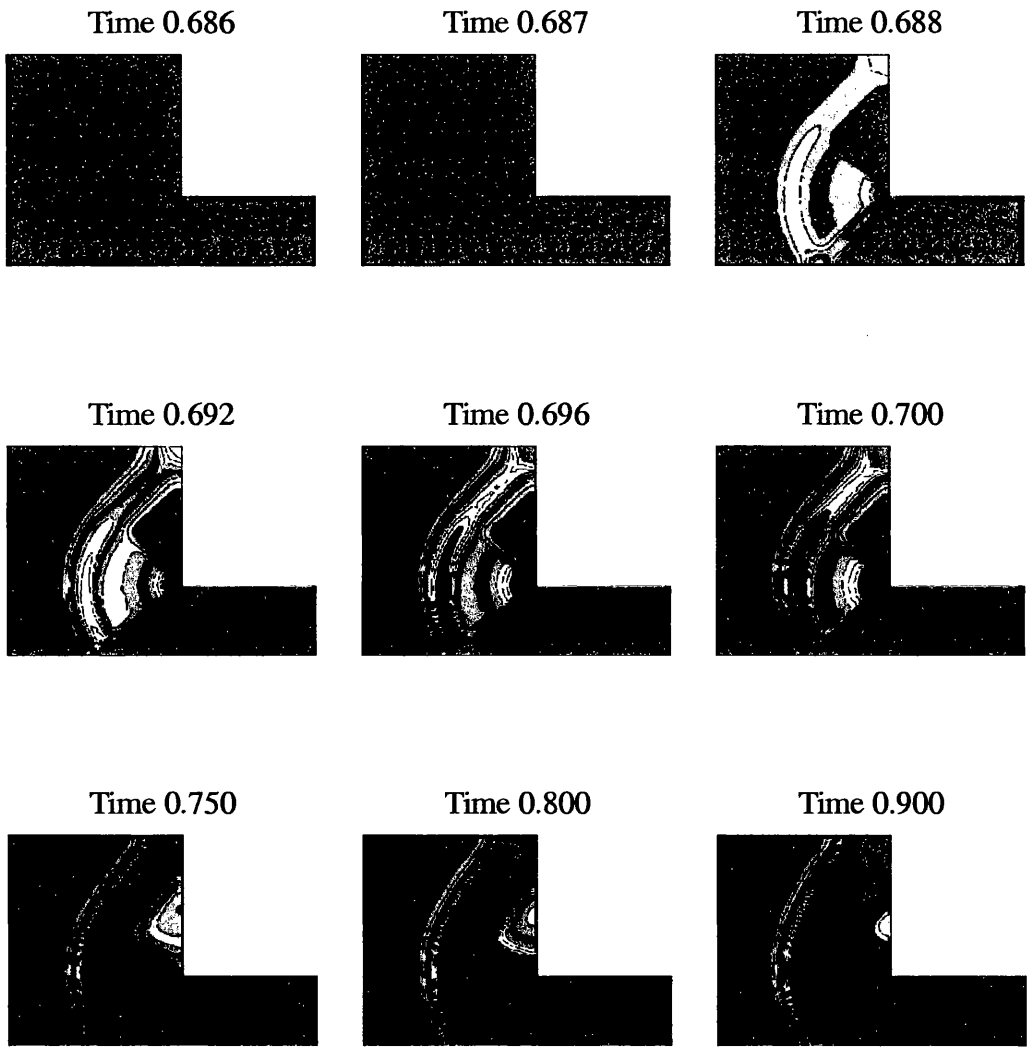


Figure 6.14: Evolution of the slip lines. The red regions represent all the values of the effective strain rate which are larger than $0.721/s$ and the blue regions represent all the values that are smaller than $0.08\ 1/s$. All the other values are in between.

6.3 Flow past a cylinder

The objective of this test was to simulate the flow of the Bingham fluid around a circular cylinder with a particular focus on the form of slip line field. For this purpose a Bingham fluid with a very small value for the plastic viscosity is used, similar to example 6.2. Apart from the slip line field the value of the *extrusion pressure* is also of interest.

This *extrusion pressure* will be compared against a value that is obtained by performing simulations that employ an elasto-plastic material. The reason for using this method is the lack of an analytical solution. The same analogy that was used for the 2D ram extrusion test will be employed.

The geometry and the boundary conditions for the elasto plastic problem are shown in Figure 6.15.

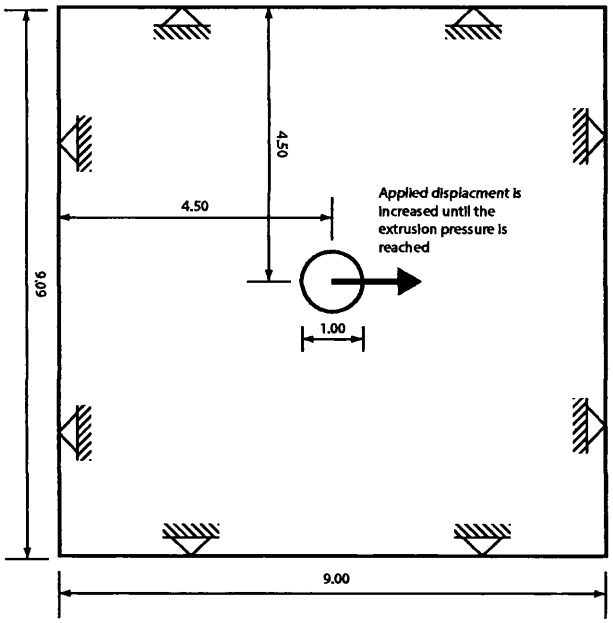


Figure 6.15: Geometry and the boundary conditions for the elasto-plastic problem

The mesh that was used had 9734 elements and 9928 nodes. It can be seen in Figure 6.16. Linear quadrilateral elements were used to simulate the elasto-plastic material. To model the elasto-plastic material the Tresca yield criterion was used. The yield stress is the same as for the Bingham material. The Tresca material properties can be found in Table 6.6.

The chosen geometry is the same as in the standard test for a flow of Newtonian fluid around the cylinder, (Simo & Armero, 1994). This geometry is represented in Figure 6.17. The properties of the Bingham fluid are given in Table 6.7.

A finite element mesh shown in Figure 6.18 has been employed. The mesh consists of 9848 elements and 5061 nodes. The mesh has been refined around the

UNIVERSITY OF WALES SWANSEA LIBRARY

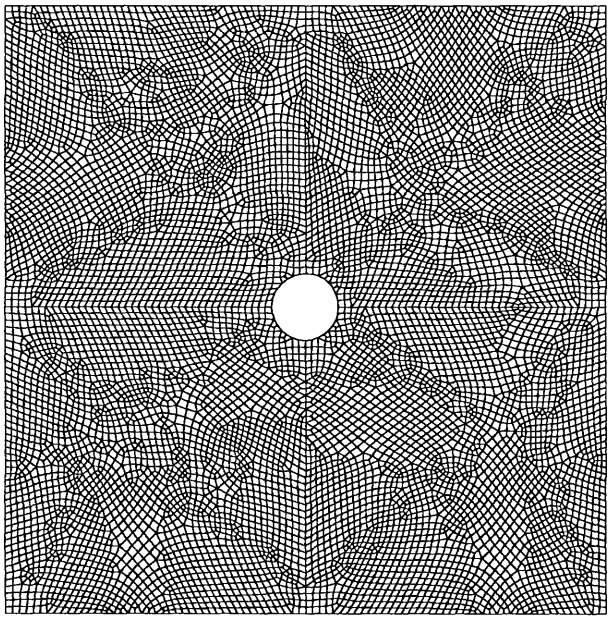


Figure 6.16: Mesh for the elasto-plastic problem

Elasto-plastic model name	Tresca
Young modulus $E[Pa/m^2]$	1E8
Poisson's ratio $\nu[]$	0.49
Density $\rho[kg/m^3]$	10.0
Yield stress $\sigma_Y[Pa/m^2]$	1000.0

Table 6.6: Tresca elasto-plastic properties

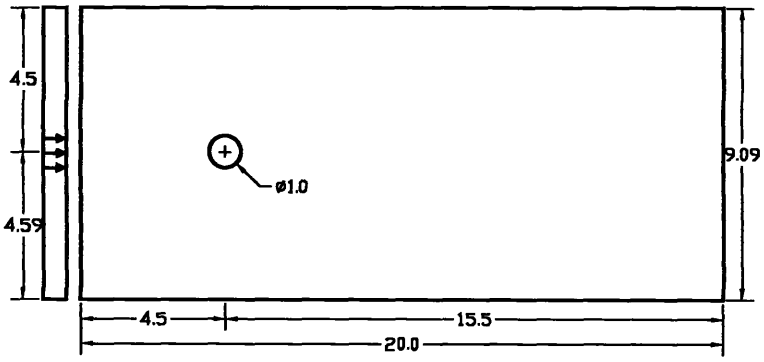


Figure 6.17: Geometry

cylinder because of the expected high gradients in the solution variable. The simulation is performed by employing 60 increments. One of the reasons for such a high number of increments is that the value of the *extrusion pressure* is not known in advance. The other reason is that adequate resolution for extrusion pressure is

Fluid properties

Fluid model name	Bingham plastic
Plastic viscosity $\mu[Pa \cdot s]$	1.0
Yield Stress $\sigma_Y[Pa]$	1000.0
Numerical properties	
Stress growth exponent[3] $m[-]$	1000000.0

see JNNFM (Blackery & Mitsoulis, 1997)

Table 6.7:

needed. An automated time integration scheme could used to reduce the number increments. This has not been done because of convenience.

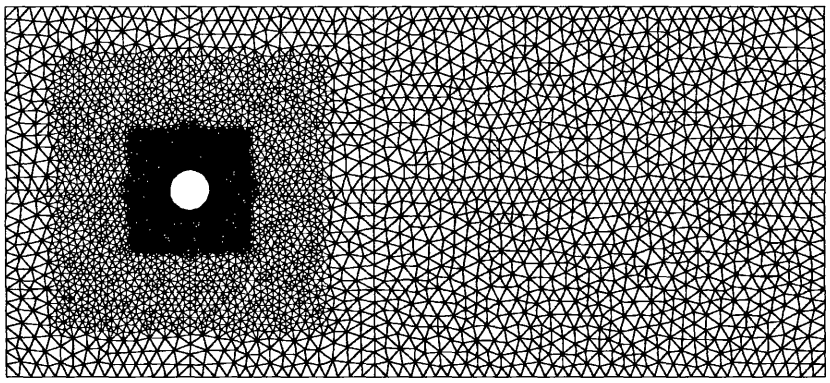


Figure 6.18: Mesh for the cylinder example

Euclidean norm

ITERATION	$t = 86.0s$ $\Delta t = 1.000s$	$t = 86.274s$ $\Delta t = 0.002s$	$t = 101.150s$ $\Delta t = 0.010s$	$t = 118.491s$ $\Delta t = 1.010s$
1	0.229376	0.328148	0.261727E-01	0.648347E-03
2	0.204081	0.143492	0.448398E-03	0.993403E-05
3	0.159021	0.945603E-02	0.764222E-05	
4	0.440011E-01	0.487602E-04		
5	0.270798E-02	0.128232E-08		
6	0.746893E-05			
7	0.447552E-10			

Table 6.8: Convergence table

The convergence for several time steps can be observed in Table 6.8. As with the previous tests quadratic convergence has been obtained. A comparison with

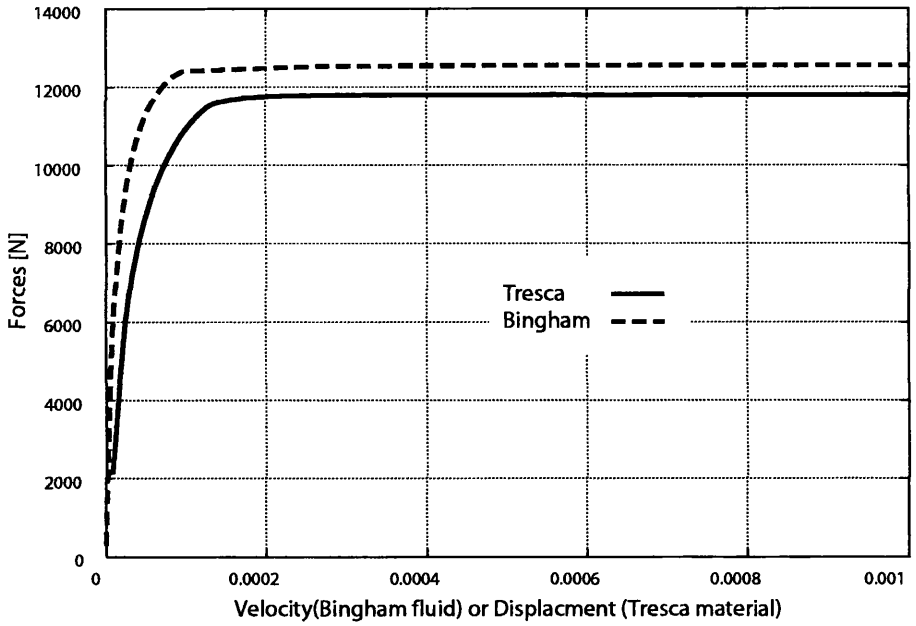


Figure 6.19: Comparison with elasto-plastic solution

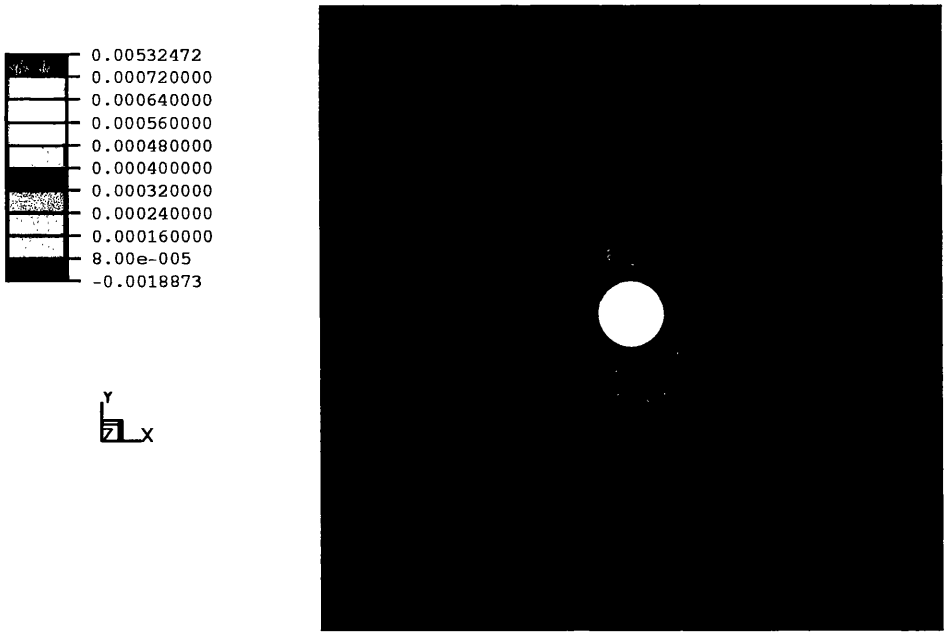


Figure 6.20: Slip lines for the elasto-plastic model

the elasto plastic solution can be found in Figure 6.19. The discrepancies could be reduced if a finer mesh was used for the elasto-plastic model. The evolution of the slip lines for the Bingham fluid model is represented in Figure 6.21. The first three pictures show the state before extrusion pressure is reached; the rest show the

state after the extrusion pressure has been reached. It can be observed that there is a gradual build up of effective strain rate at the front and back of the cylinder before the extrusion pressure is reached. The slip lines that are observed start by being symmetrical about the vertical axis but later (time=88.9784) move to the rear of the cylinder. Slip lines obtained for the elasto-plastic model in Figure 6.20 correspond well to the ones obtained for the Bingham fluid model.

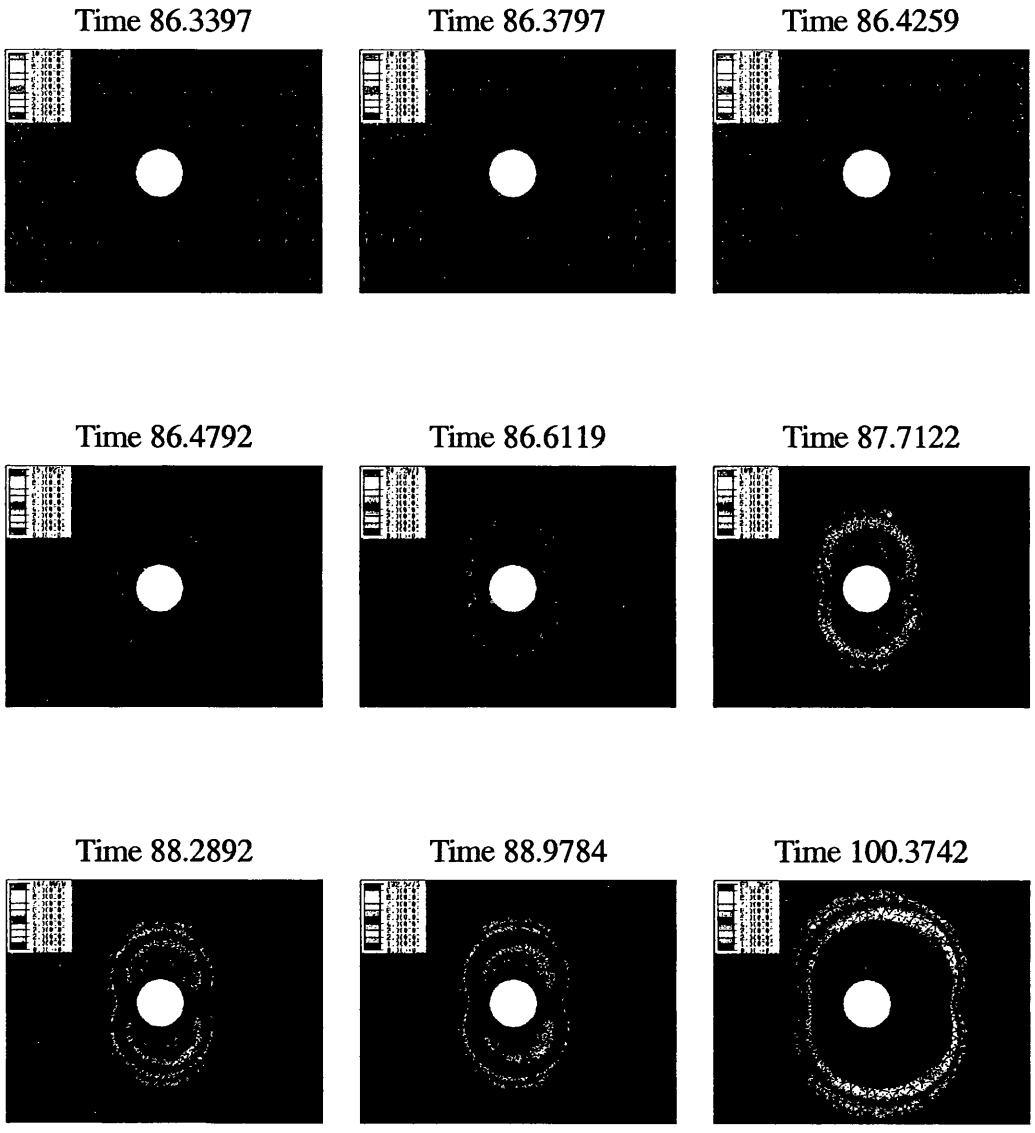
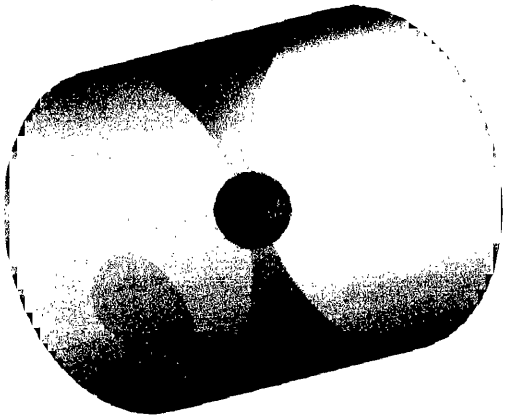


Figure 6.21: Evolution of the slip lines

6.4 Flow past a sphere

This 3D test involves a sphere in a Bingham fluid as shown in Figure 6.22(a). It was done in order to verify the 3D implementation. The properties of the fluid are given in Figure 6.23(a). The test involves a rigid sphere that is immersed in a cylinder of fluid. The geometry of the problem shown in Figure 6.22(b). The pressure is applied at one of the bases of the cylinder. This is modelled using a traction boundary condition. The test involves increasing the pressure on the base of the cylinder until the yield stress is reached and the fluid starts to flow. The value of pressure at which the fluid starts to flow is compared to a value that is obtained by modelling this problem using an elasto-plastic material. To model the elasto-plastic material an elasto-plastic material with the Tresca yield criterion is used. The properties are given in Table 6.9.

a.) Model



b.) Dimensions

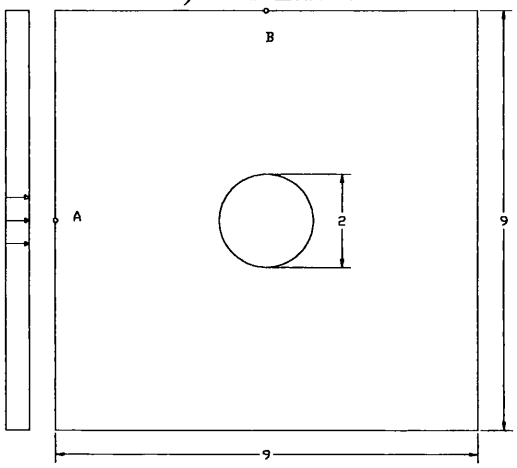


Figure 6.22: Geometry for the 3D sphere example

a.) Fluid properties

Fluid model name	Bingham plastic
Plastic viscosity $\mu[Pa \cdot s]$	0.1
Yield Stress $\sigma_Y[Pa]$	1000.0
Density $\rho[kg/m^3]$	10.0
Numerical properties	
Stress growth exponent[4] $m[-]$	1000.0

see JNNFM (Blackery & Mitsoulis, 1997)

b.) Ramp load function

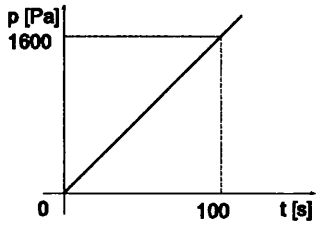


Figure 6.23: a.) Table of fluid properties b.) Ram load function

A finite element mesh shown in Figure 6.24 has been employed in the simulation. This mesh consists of 43628 elements and 8088 nodes. The simulation

Elasto-plastic model name	Tresca
Young modulus $E[Pa/m^2]$	1E8
Poisson's ratio $\nu[]$	0.49
Density $\rho[kg/m^3]$	10.0
Yield stress $\sigma_Y[Pa/m^2]$	1000.0

Table 6.9: Tresca elasto-plastic properties

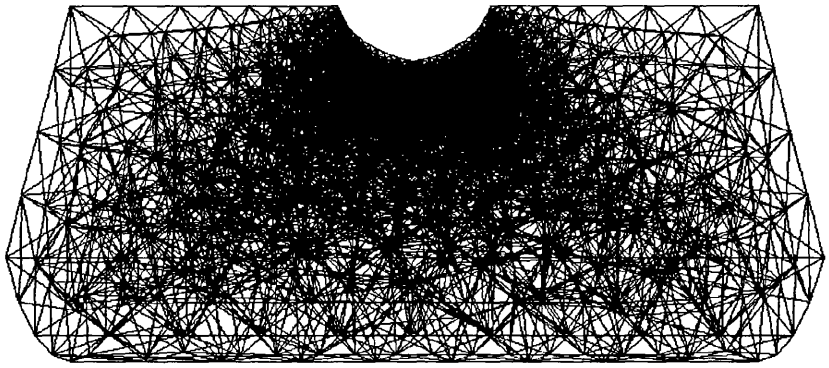


Figure 6.24: Mesh for the sphere example

is performed by employing 80 increments. The number of increments have been reduced compared to the previous example by designing the time stepping scheme. This has been done by using the information gained from the elasto-plastic simulation. The convergence for several typical time steps can be observed in Table 6.10.

	Euclidean norm			
ITERATION	$t = 10.0s$ $\Delta t = 10.0s$	$t = 46.0s$ $\Delta t = 1.0s$	$t = 48.5s$ $\Delta t = 0.1s$	$t = 50.0s$ $\Delta t = 0.1s$
1	1.44691	0.286617	0.114772	0.168495
2	0.163656	0.252209E-01	0.835202E-02	0.371211E-02
3	0.188853E-02	0.373172E-03	0.360095E-04	0.210655E-05
4	0.188119E-06	0.673452E-07	0.546683E-09	0.356581E-09
5	0.235252E-12	0.149580E-11	0.454517E-11	

Table 6.10: Convergence table

To verify the results obtained by performing the flow simulation of the Bingham fluid a simulation is also performed for the limit load of a sphere in an elasto-plastic material. This test involved an axisymmetric sphere that is immersed in an elasto-plastic cylinder, using the Tresca yield criteria. The finite element mesh that

is used is shown in Figure 6.26, and consists of 4618 elements and 14107 nodes. After the numerical tests were performed the values of forces acting on the spheres was compared. The result of this comparison can be found in Figure 6.25. This graph shows a good comparison between the forces obtained. The nature of the evolution of the force is essentially the same.

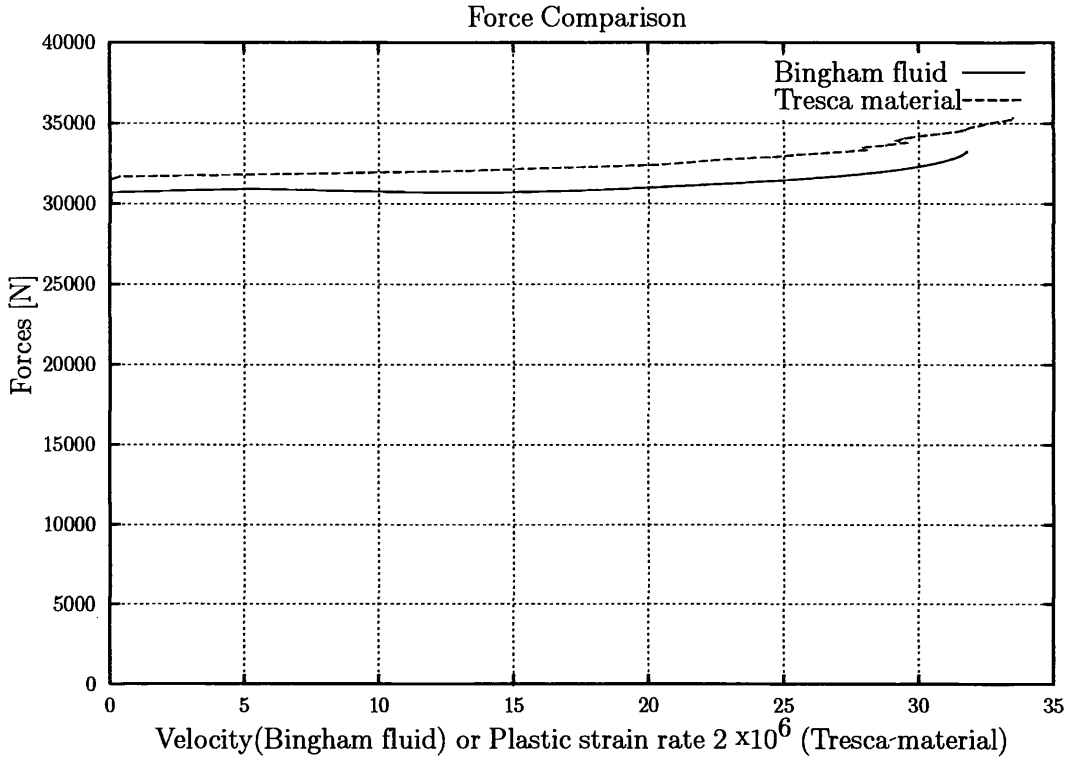


Figure 6.25: Comparison between forces obtained using 3D Bingham model and using a 2D axisymmetric elasto-plastic model

Figure 6.27 shows the evolution of the slip line field around the the sphere. The figure shows the axial cross-section. The slip lines that are obtained are similar in there qualitative appearance to the previous tests. The shape is similar to the ones that were obtained by Beris *et al.* (1985).

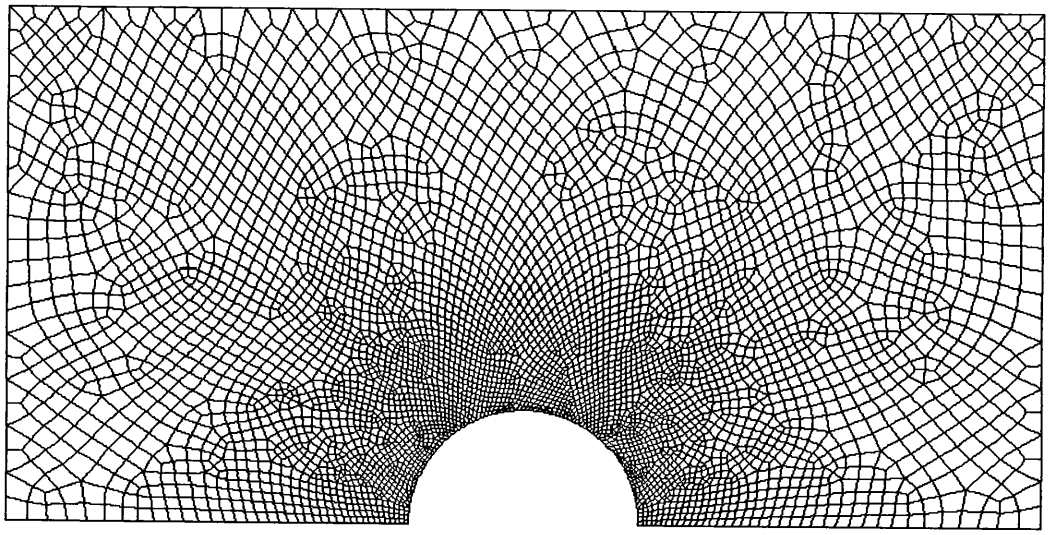


Figure 6.26: Mesh for the 2D axisymmetric sphere

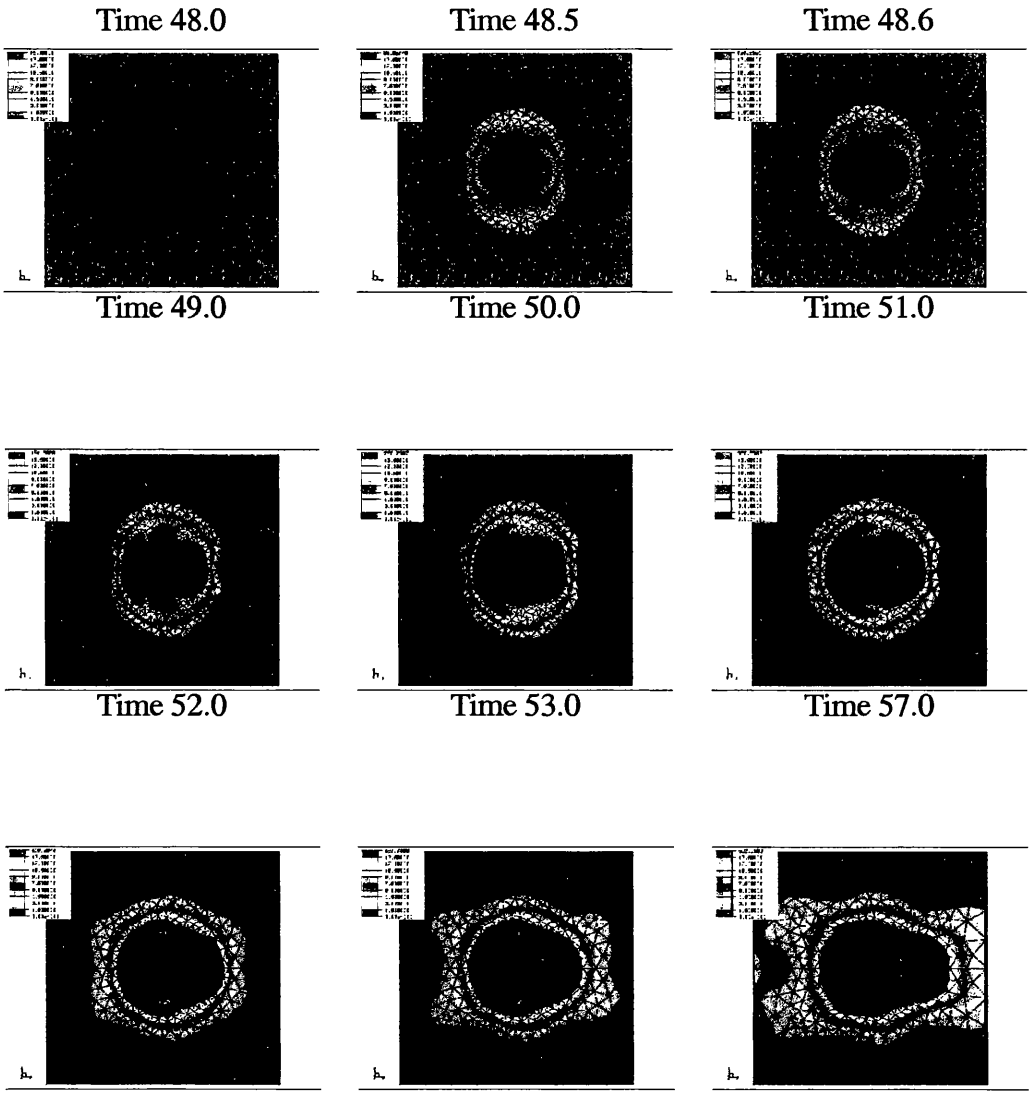


Figure 6.27: Evolution of the slip line field for a flow of the Bingham fluid around a sphere

6.5 3-D flow in a single screw extruder channel

The objective of this example is to gain a better understanding of the steady-state non-Newtonian fluid flow over a tangential self-wiping, single screw extruder. A horizontal cross-section through such an extruder is shown in Figure 6.28. The non-Newtonian fluid is modelled using the power law model. The flow is assumed to be isothermal with no-slip conditions on the boundaries. The simulation also illustrates the use of particle tracking to track the movement of material particles inside the screw channel.

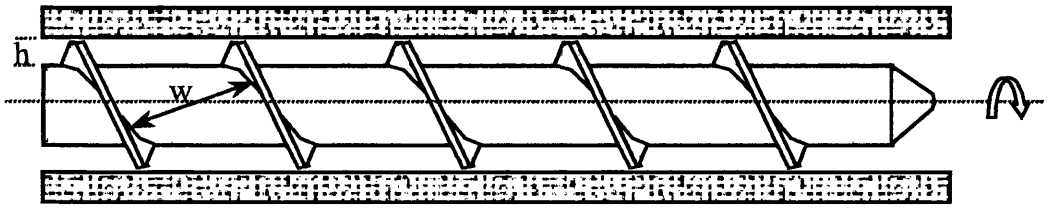


Figure 6.28: Horizontal cross section of a single screw extruder

The flow inside the extruder is modelled by using the so-called "moving barrel formulation" approach. The "moving barrel formulation" is based on the screw channel being unwound. This modelling approach is valid if the velocity field does not change significantly along the channel direction (Sastrohartono & Kwon, 1990). Also the screw channel is assumed stationary while the barrel housing moves over it in the opposite direction to the screw rotation. The angle of the barrel housing in relationship to the channel is equal to the pitch angle θ of the screw. The geometry of the unwound screw channel, the velocity components and the coordinate system for the flow analysis are shown in Figure 6.29.

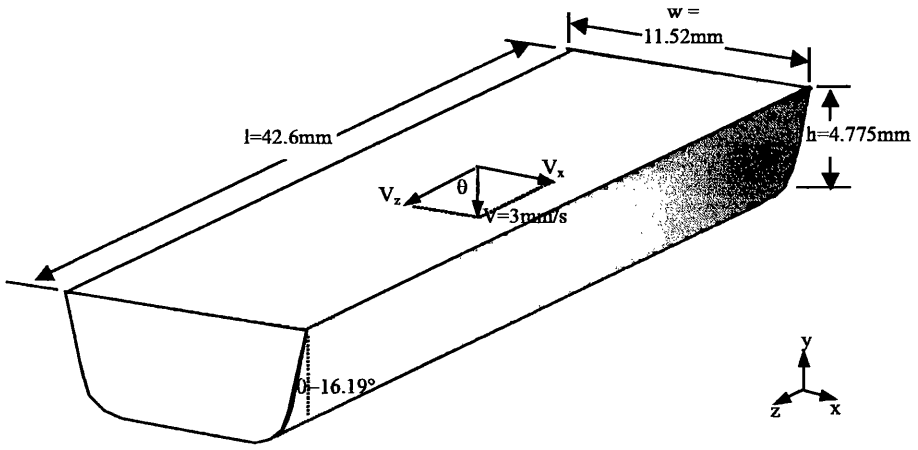


Figure 6.29: Dimension of the channel

The velocity components on the surface of the channel are fixed except the

inflow and outflow surface. The top surface is assigned the velocity of the moving barrel. On the screw surface the velocity components are equal to zero. This boundary condition is also applied to edge where the barrel and screw surface meet. If the boundary conditions on the edge were equal to barrel surface conditions leakage would occur. The moving barrel velocity is projected to give a velocity component along channel axis and a component orthogonal to it. Details can be found in Figure 6.30.

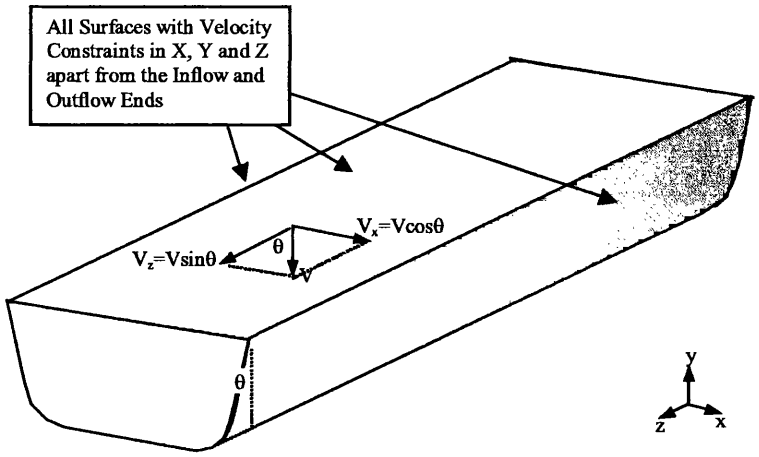


Figure 6.30: Boundary conditions

As with all previous examples an unstructured mesh has been adopted with a default element size of 0.8 mm. The mesh was refined along the top edges of the channel with a element size of 0.4 mm. This setup yields a mesh with 34296 elements and 8030 points (24120 dof). An isometric view of the mesh with the mesh size details can be found in Figure 6.31. This numerical test is substantially larger than the previous tests which makes it useful in demonstrating robustness of the numerical scheme.

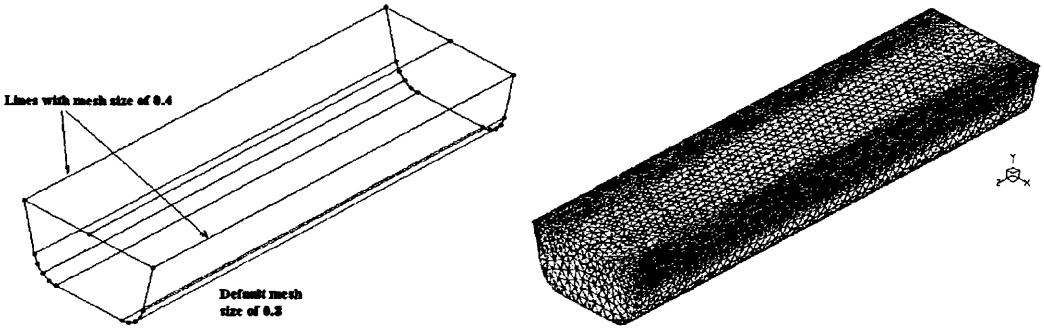


Figure 6.31: 3D Mesh and specified mesh sizes

A power law model was used to describe the material behaviour. The power law parameters can be found in Table 6.11. The first three are standard power law

Property	Value	Units
ρ	1.0	kg/mm^3
m	0.04709	$KPa \cdot s^n$
n	0.31	$-$
$\dot{\gamma}_c$	0.1 or 0.001	s^{-1}

Table 6.11: Values for the fluid parameters

parameters the last parameter is a numerical one. The parameter $\dot{\gamma}_c$ is used to decide where to switch between a power law model and linear one as $\dot{\gamma}_c \to 0$. The reason that this is necessary is that the power law model is not defined for $\dot{\gamma} = 0$ see Section 3.2.1. The modified power law model has the following form

$$\mu(\dot{\gamma}) = \begin{cases} m\dot{\gamma}^{n-1}, & \dot{\gamma} \geq \dot{\gamma}_c \\ \mu'(\dot{\gamma} - \dot{\gamma}_c) + \mu_c, & \dot{\gamma} < \dot{\gamma}_c \end{cases}$$
$$\mu' = m(n-1)\dot{\gamma}^{n-2}, \quad \mu_c = m\dot{\gamma}_c^{n-1}$$

where m is the fluid consistency index, n is the power law index and $\dot{\gamma}_c$ is the critical shear rate. Two different critical shear rates were chosen 0.1 and 0.001. This was done to evaluate how appropriate the chosen values are. The chosen value of the critical shear rate in general should be lower than the values of shear rate that occur in the numerical problem. As the critical shear rate decreases convergence is more difficult to archive and the time step needs to be reduced.

The two values of the critical shear rate give similar velocity fields that can be observed in Figures 6.34 and 6.35. This corresponds to the predictions since most of the domain has shear rate that are higher than the critical shear rate. The strain rate distribution can be observed in Figure 6.36.

The velocity field given in Figure 6.34, shows a strong circular movement to one side of the channel. This is in correspondence with expectation. By viewing the particle paths that can be found in Figures 6.37, 6.38 and 6.39 the three dimensional nature of the velocity fields can be observed. The circular movement is reiterated in Figure 6.38. The other two particle path figures show how particle are transported through the channel of the screw. This provides useful information when choosing or designing an extrusion screw for a particular material.

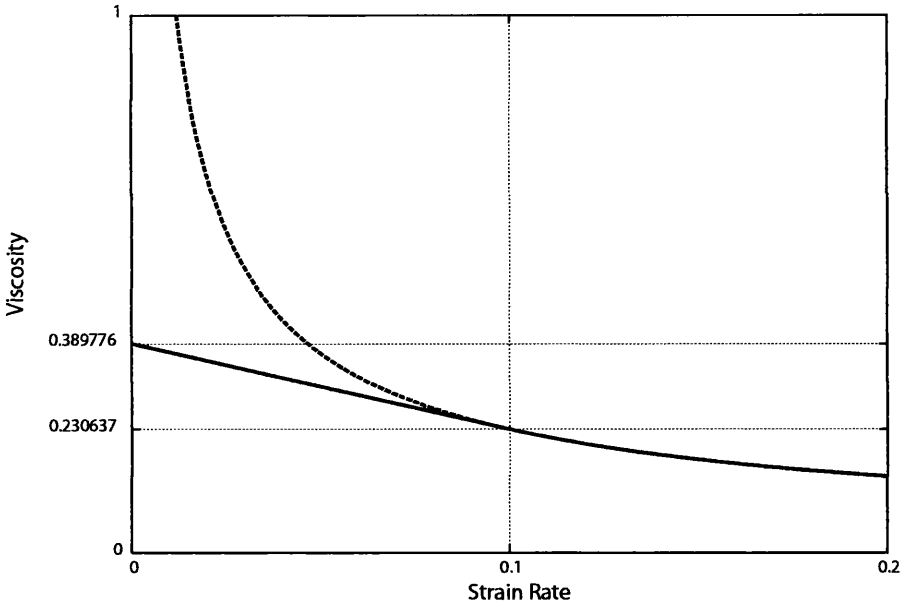


Figure 6.32: Plot of the material law for $\gamma_c = 0.1$

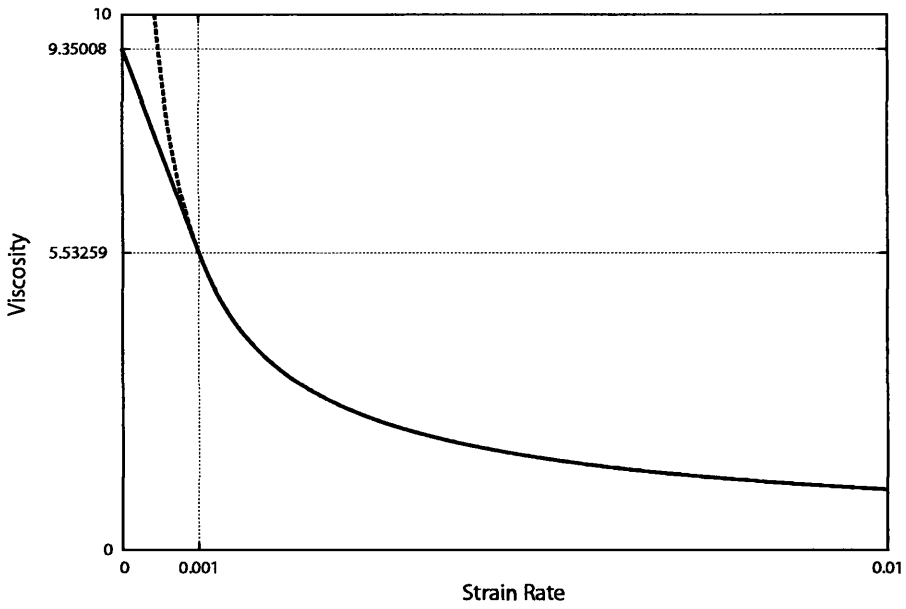


Figure 6.33: Plot of the material law for $\gamma_c = 0.001$

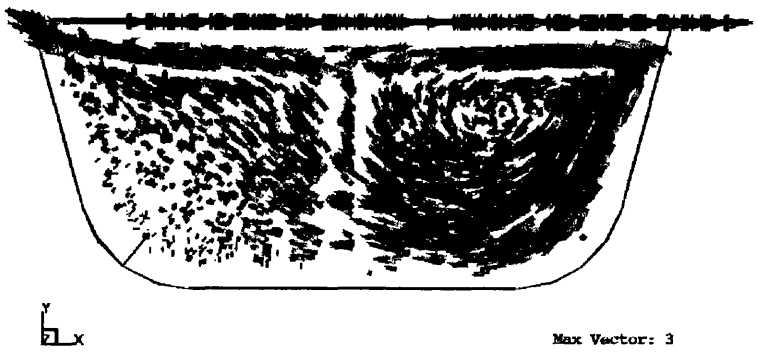


Figure 6.34: Velocity field for $\gamma_c = 0.1$

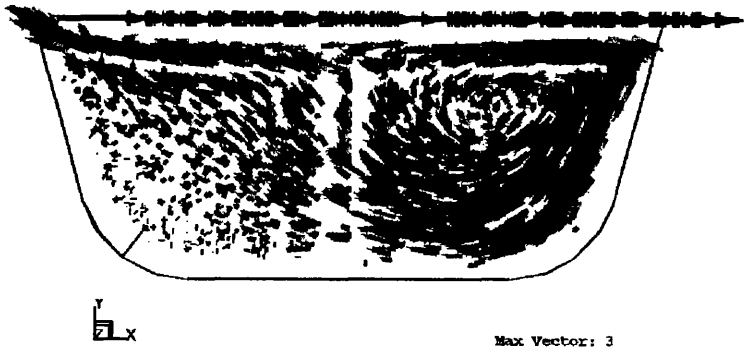


Figure 6.35: Velocity field for $\gamma_c = 0.001$

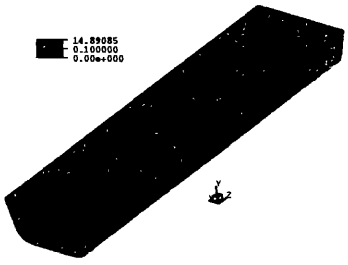


Figure 6.36: Strain rate distribution

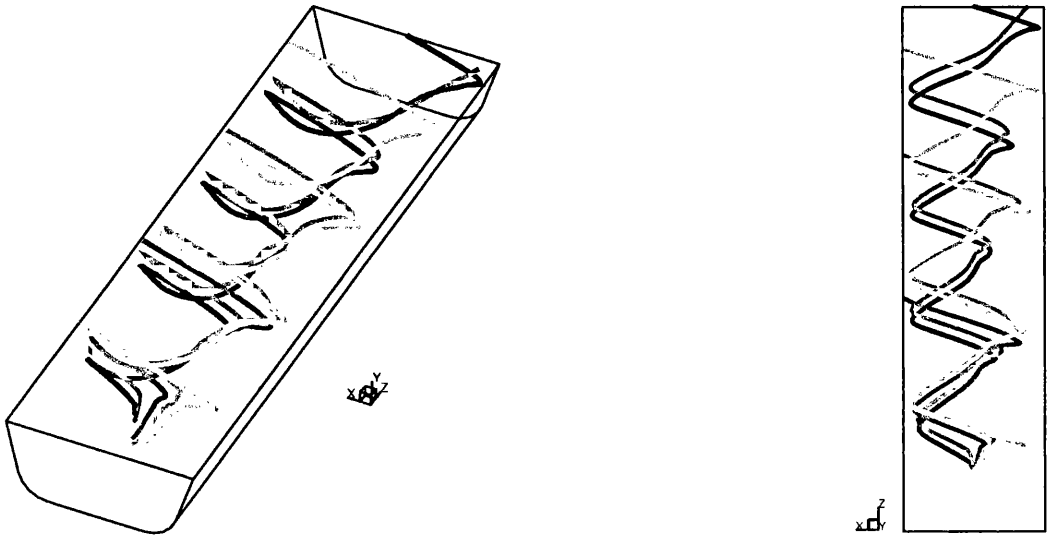


Figure 6.37: Particle paths

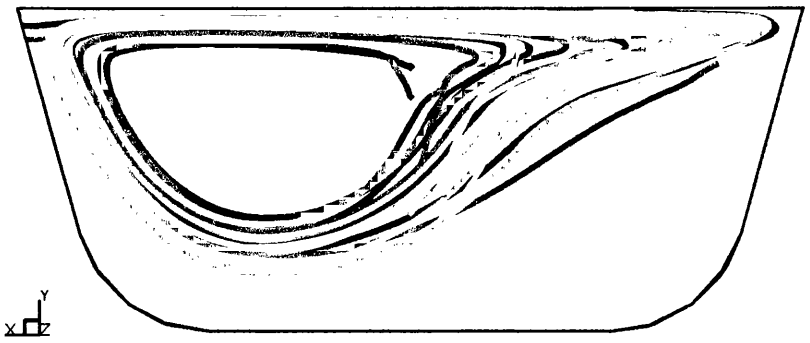


Figure 6.38: Particle paths - XY plane

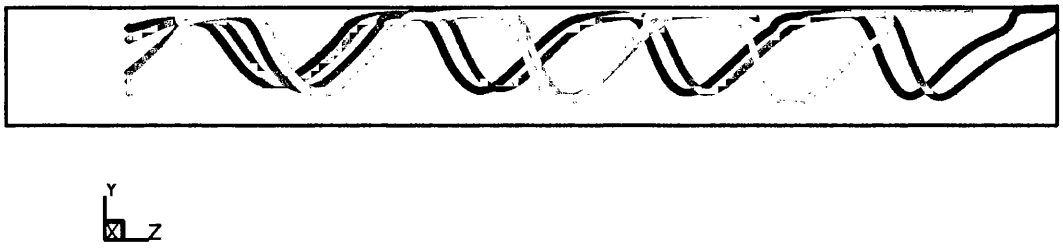


Figure 6.39: Particle paths - YZ plane

6.6 3-D coupled simulation of die and extrusion flows

The objective of this simulation is to model the coupling of die and extrusion flows. The material is polymeric and is modelled by power law model. The flow is assumed to be steady-state and isothermal with no-slip conditions on the die and mandrel boundaries.

The model is a die and extruder combination where the complex annular die is preceded by a co-rotating fully inter meshing self-wiping twin screw extruder, see Figure 6.40. The approach region of the die is 100 mm in length and has a figure 8 cross-section. The entrance expands onto a star-shaped mandrel via a 10 mm connector plate. The star-shaped mandrel has an 11-point insert and an outer diameter of 61 mm. The average inner diameter is 30 mm. The two 30 mm rotating shafts of the twin screw extruder protrude into the approach region of the die and act as screw inserts. Pictures that schematically illustrate this problem can be found in Figure 6.40.

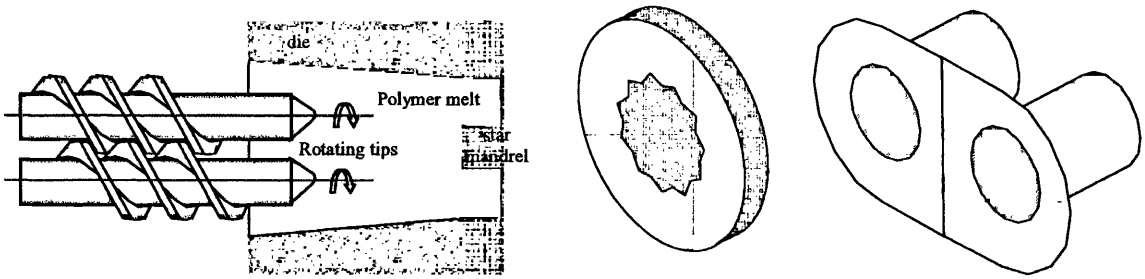


Figure 6.40: Problem description

The problem is modelled by using a 3-D steady-state flow analysis which fully takes into account the rotation of the screw tips in the die flow. A diagram that shows the discretized model together with its dimensions can be found in Figure 6.41.

As had been already mentioned the power law material model is used. The values for the fluid consistency index m, power law index n, density ρ and critical shear rate $\dot{\gamma}_c$ that where used to model can be found in Table 6.12. Details about the power law model can be found in section 3.2.1.

Property	Value	Units
ρ	1.0	kg/mm^3
m	0.04709	$KPa \cdot s^n$
n	0.31	–
$\dot{\gamma}_c$	0.1 or 0.0003	s^{-1}

Table 6.12: Values for the fluid parameters

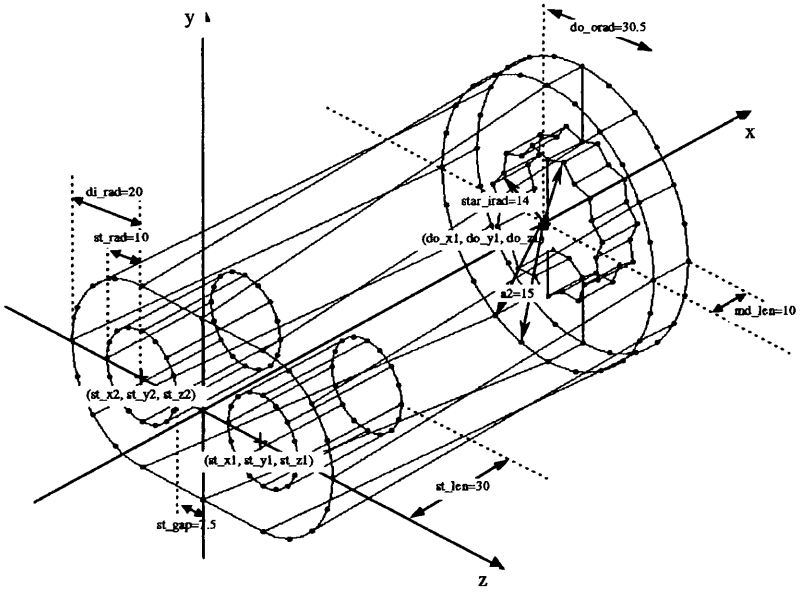


Figure 6.41: Problem dimensions

As with the previous example two different values for critical shear rate were chosen. It will be shown that the lower shear rate is the appropriate one to use with this problem and these boundary conditions. The model needs to adequately represent the extrusion process. The process involves a rotational velocity in the inflow. This loading condition is applied by using a cylindrical coordinate system, which is then transformed into the standard Cartesian coordinate system. The velocities are fixed on surfaces except on the connector plate and the outflow surface. Details of this can be found on Figure 6.42.

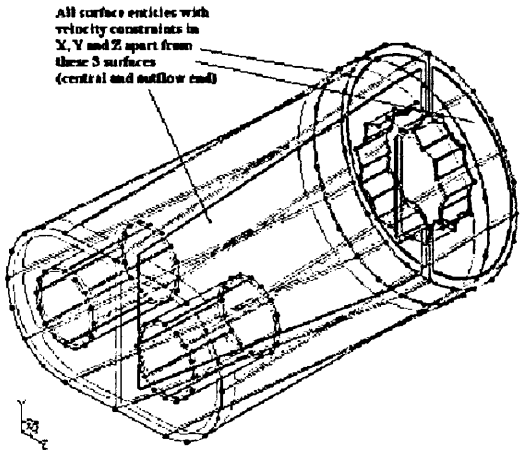


Figure 6.42: Boundary condition

The velocities on the inflow surface have a prescribed inflow velocity of

$0.05mm/s$. A rotational velocity is prescribed to the immersed rotating shafts. This rotational velocity is equal to $0.09mm/s$. The constant inflow velocity could be replaced by a velocity profile from an extrusion simulation or a traction boundary condition could be applied. These different boundary conditions all try to replace the the screw extruder. Ideally the die and extruder should be modelled together. In some situations this is paramount and the two can not be separated. This is especially necessary when a breaker plate is not used or the screws feeding the die are situated close to the entry of the die (Lawal *et al.*, 1996). In this case only the immersed rotating shafts are taken into an account. The loading conditions are illustrated in Figure 6.43.

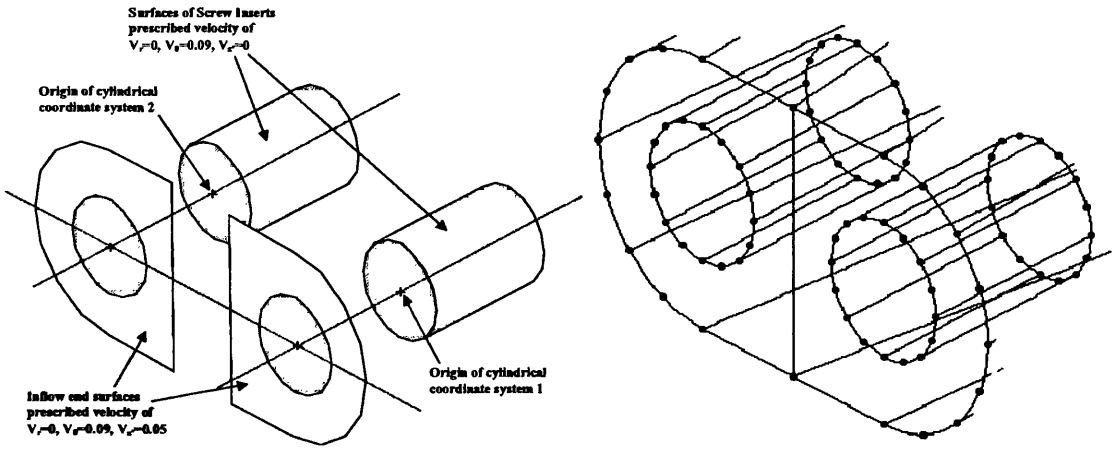


Figure 6.43: Loading conditions

To discretize the above model an unstructured mesh was used. The default element size was set to 5 mm. The element order as with all the other problems is linear. With these parameters the mesh generator produced the mesh shown in Figure 6.44. The mesh consists of 2999 nodes and 14715 elements.

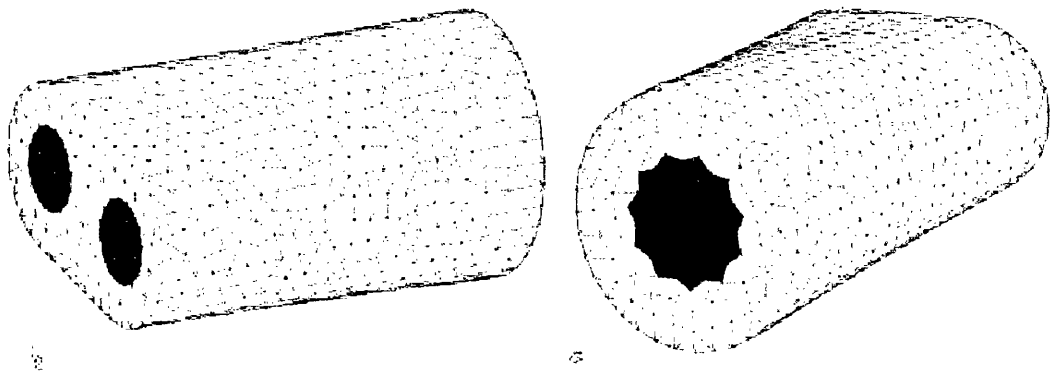


Figure 6.44: Mesh

6.6.1 Results

The results obtained after running the steady state problem for the two critical shear rate values can be observed in Figures 6.45, 6.46, 6.47 and 6.48. It can be observed that there is a significant difference in the results for the two critical shear rate values. This means that lower value of shear rate needs to be used. Using a lower value of the critical shear rate than $\dot{\gamma} = 0.0003$ would make no difference but would introduce numerical difficulties. The reason that a lower value would not make any difference to the solution can be justified by observing the lowest value for the shear rate in Figure 6.46. The lowest value for the shear rate is $\dot{\gamma} = 0.000360377$ and this higher than the chosen critical shear rate $\dot{\gamma}_c = 0.0003$. The contour plots in Figures 6.45 and 6.47 represent a quantitative insight of the velocity and pressure fields. The velocity is highest between the two shafts. With the pressure contour plots the pressure gradient that drives the flow can be observed.

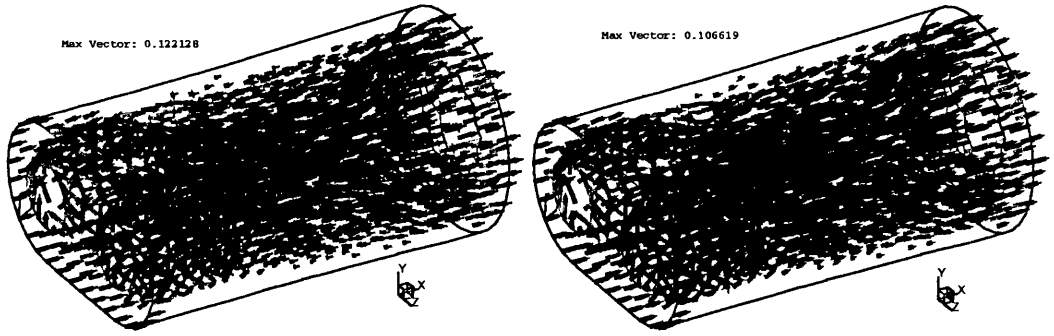


Figure 6.45: Velocity field for $\dot{\gamma}_c = 0.1$ (left) and $\dot{\gamma}_c = 0.0003$ (right)

Figure 6.49 represents the particle path of a group of preselected particles. This type of post processing further improves the understanding of the flow that occurs in this extruder and die combination. The particle path figures illustrate the straightening of the fluid flow that occurs in the die approach region. The better understanding of fluid flow helps to improve the die design.

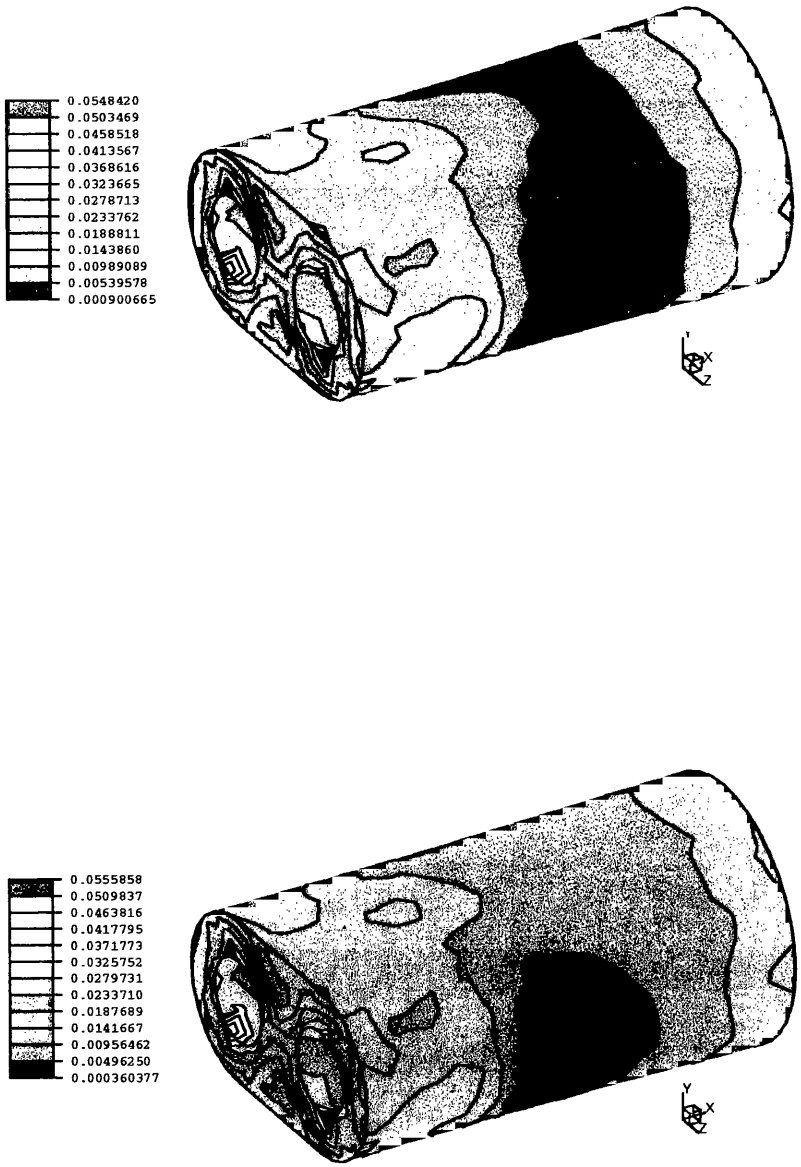


Figure 6.46: Strain rate for $\dot{\gamma}_c = 0.1$ (top) and $\dot{\gamma}_c = 0.0003$ (bottom)

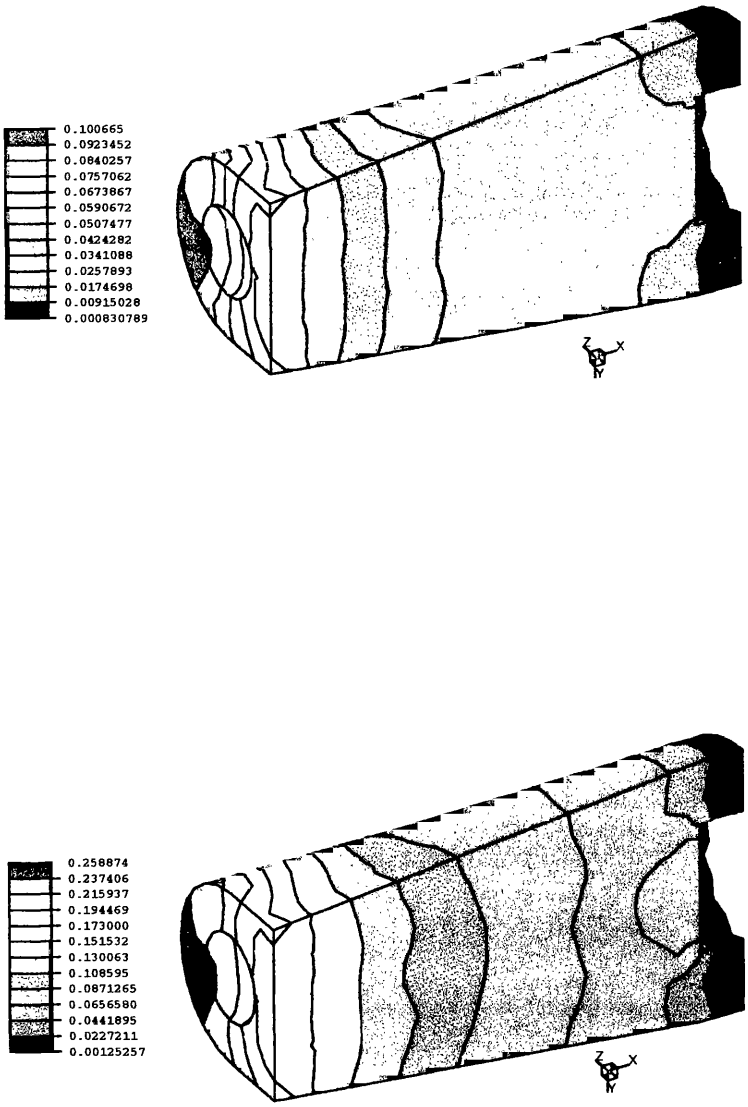


Figure 6.47: Pressure field for $\dot{\gamma}_c = 0.1$ (top) and $\dot{\gamma}_c = 0.0003$ (bottom)

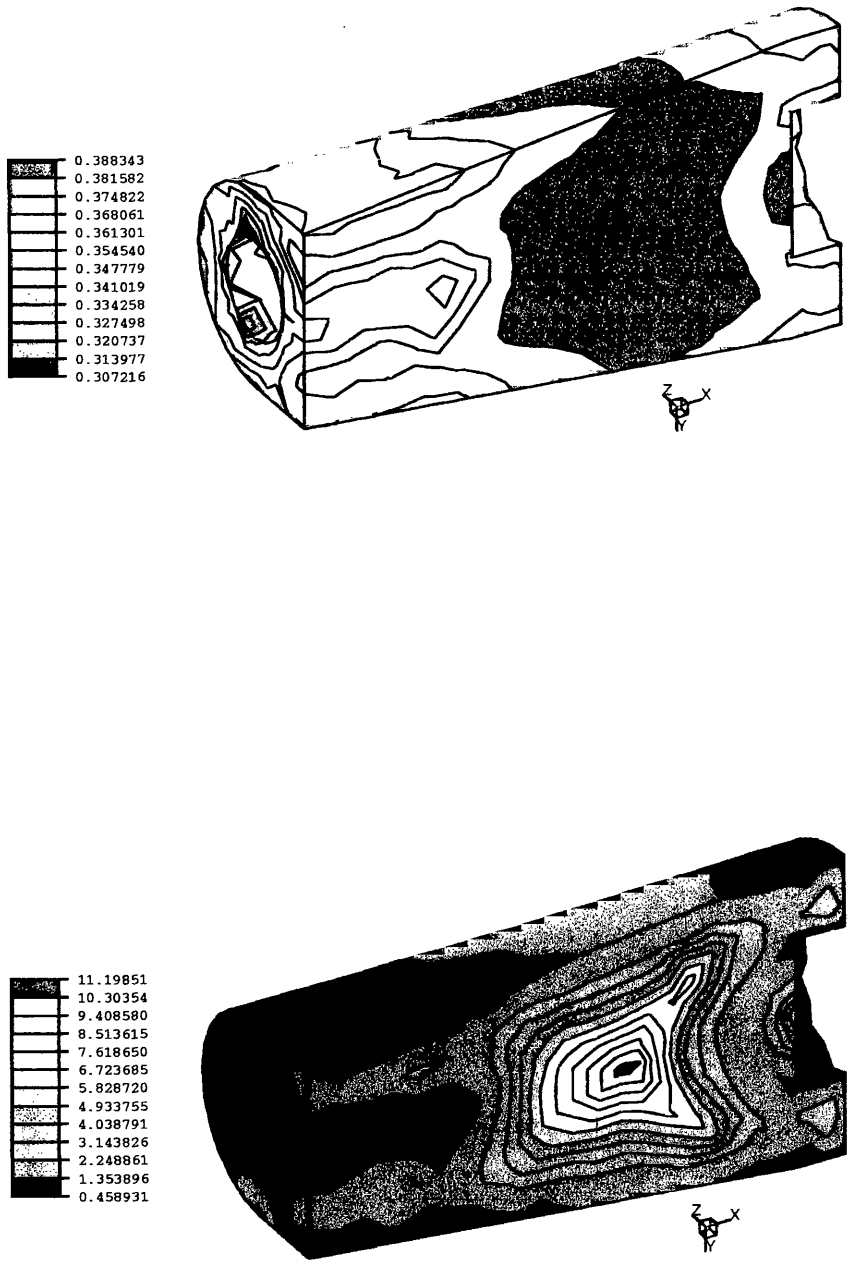


Figure 6.48: Viscosity distribution for $\dot{\gamma}_c = 0.1$ (top) and $\dot{\gamma}_c = 0.0003$ (bottom)

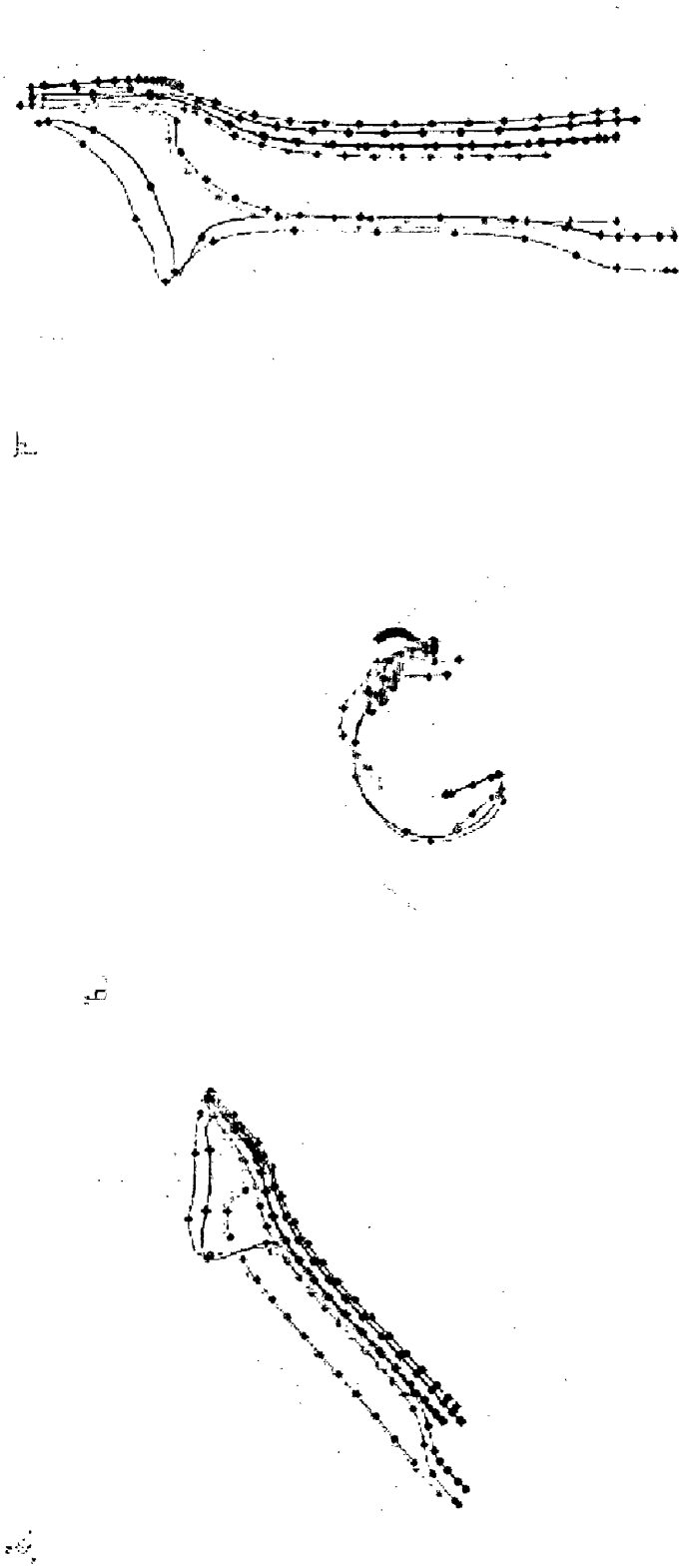


Figure 6.49: Particle paths

Chapter 7

Conclusions

In this thesis an implicit semi discrete finite element formulation for non-Newtonian fluids has been successfully developed and implemented into a finite element program. The formulation builds on the success of the stabilised finite element method for Newtonian fluids, that enhances numerical stability without compromising consistency (Franca, 2001). In the first three chapters an introduction to non-Newtonian fluid mechanics has been described. Chapter 4 apart from introducing stabilised finite element methods and problem associated with numerical modelling provided a formulation that fully describes the behaviour of non-Newtonian fluids. The discretization was achieved by using linear equal order interpolation functions for velocity and pressure. The stability problems associated with using equal order interpolation functions with mixed methods were overcome by circumventing the Babuška–Brezzi condition. This makes the implementation much easier than using for instance tri–quadratic and tri–linear interpolation functions for velocity and pressure respectively as in Alexandrou *et al.* (2001). In the next chapter the numerical procedure that is used to solve the resulting evolution problem has been elaborated. The highly non linear nature of the problem was emphasised together with the need for iterative solution procedure. The Newton-Raphson iterative solution procedure has been chosen for this purpose. The procedure has rapid convergence that is superior when compared to procedures that involve successive substitutions (Liu *et al.*, 2002). It also eliminates the need for pressure correction as in Baloch *et al.* (1995). For the type of problem considered in this thesis the Newton-Raphson solution procedure is robust and efficient but requires linearization of governing equations that yields a consistent tangent stiffness matrix. This is usually quite difficult to achieve for complex problems such as the one considered here. Often the tangent stiffness matrix is calculated approximately by numerically differentiating the residual vector. This does not give satisfactory results. Therefore an analytical value for tangent stiffness matrix has been derived. Once the analytical form had been obtained the implementation had been automated, by using the "Mathematica" package. The use of "Mathematica" yielded a general framework that can be easily applied to new formulations. To verify the implementation several numerical tests

were performed.

The tests showed that the implementation was executed correctly and sufficient accuracy has been achieved. The tests used various methods to verify accuracy. First comparison with analytical solutions was performed. This compared nodal values for a 2D Couette flow. Secondly the extrusion pressure in a 2D extrusion test compared well to the extrusion pressure that was obtained using the slip line theory. Thirdly a 3D test that uses the analogy between a Bingham fluid material with low values for the plastic viscosity and an elasto-plastic material with Tresca yield criterion, showed remarkably good correlation between the results. It confirmed that the use of Papanastasiou (1987) Bingham fluid model for numerical modelling gives satisfactory results (Smyrnaios & Tsamopoulos, 2001).

All this shows that numerical modelling of non-Newtonian fluids via the proposed implicit finite element method gives very good results with very good accuracy. A number of the implementation strategies that were used in finite element formulation came from modelling of Newtonian fluids. It has been shown that concepts like SUPG, PSPG and one step discrete time stepping schemes were successfully transferred from Newtonian fluids to non-Newtonian fluid mechanics and gave good results.

By employing the Newton-Raphson solution technique in the context of implicit time stepping schemes a robust and fast solution procedure has been obtained. It proved exceptionally robust when dealing with rapid changes in the solution. An extreme example of such a situation can be observed in the extrusion example where the velocity changes several orders of magnitude in a single time step. This is specially prominent when the extrusion pressure is reached. All the tests showed quadratic convergence once the solution fell within the "ball of convergence". This can be observed in relevant tables that feature the Euclidean norm for several characteristic time steps.

It has been shown that the method can be used to model industrial processes. This can be observed in the last two numerical examples that model single and twin screw extruders.

7.1 Recommendation for future work

As a test of robustness of the proposed numerical modelling technique further simulations of large scale industrial problems could be performed. Such tests are planned and will be reported in the near future.

In future developments it would be useful to consider different time stepping schemes, such as the recently proposed generalised–α method (Jansen *et al.*, 2000). Imposing moving boundary conditions is also of interest.

It would also be useful to include more complex material models that would have a time dependent component and solid like properties, starting first with materials that have partial elastic recovery. It would be modelled by allowing the normal

stress differences to be non zero. The modelling could be taken even further by allowing viscosity to be time dependent. By adding this, phenomena like thixotropy could be modelled. The implementation of time dependent fluids would also benefit from a more elaborate time stepping scheme mentioned earlier. So implementing them together would be a useful proposition. These implementations would not require modifications in the overall computational strategy described in this thesis. Non-Newtonian fluid models that have temperature dependent viscosity could be implemented using thermal coupling.

Apart from adding new materials and time stepping schemes it would be useful to develop an adequate error estimator. This error estimator could be then used to develop adaptive mesh generation. This can in turn help to further improve the efficiency of the method and enable even larger problems to be modelled. As a first step the shear strain rate can be used as an error indicator or in the case of modelling a rigid sphere the minimum and maximum principle can be utilised (Beris *et al.*, 1985). This is simple to implement and preliminary tests have already been performed which gave satisfactory results.

Appendix A

Derivation of $\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h]$

This appendix deals with the details of the derivation of $\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h]$. The parameter τ represents the stabilising parameter. To calculate the derivative we start from the definition of τ,

$$
\begin{aligned}
Re(\mathbf{u}^h) &= \frac{\|\mathbf{u}^h\|\, h_{len}\, \rho}{2\,\mu(\mathbf{u}^h)} \\
h_{len} &= \frac{\mathrm{A}}{\pi} \quad \text{(2D problem)} \\
h_{len} &= \sqrt[3]{6\mathrm{V}} \quad \text{(3D problem)} \\
\tau &= \frac{h_{len}}{2\,\|\mathbf{u}^h\|} \frac{1}{\sqrt{1+\frac{9}{Re^2(\mathbf{u}^h)}}}
\end{aligned}
$$

The values that are used are as follows, Re is the element Reynolds number, h_{len} is the element length, A element area for 2D elements and V is the volume of 3D elements. The derivative is given as

$$
\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = \frac{d}{d\zeta}\bigg|_{\zeta=0} \tau(\mathbf{u}^h + \zeta\Delta\mathbf{u}^h) = \boldsymbol{\tau}'(\mathbf{u}^h)\cdot\Delta\mathbf{u}^h
$$

The derivative will be denoted as $\boldsymbol{\tau}'$ in the rest of the text.

$$
\begin{aligned}
\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] &= \frac{d}{d\zeta}\bigg|_{\zeta=0}\left[\frac{h_{len}}{2\,\|\mathbf{u}^h+\zeta\Delta\mathbf{u}^h\|}\frac{1}{\sqrt{1+\frac{9}{Re^2(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)}}}\right] = \\
&= \frac{d}{d\zeta}\bigg|_{\zeta=0}\frac{h_{len}}{2\sqrt{\mathbf{u}^h\cdot\mathbf{u}^h+2\,h\,\mathbf{u}^h\cdot\Delta\mathbf{u}^h+h^2\,\Delta\mathbf{u}^h\cdot\Delta\mathbf{u}^h}}\cdot\frac{1}{\sqrt{1+\frac{9}{Re^2(\mathbf{u}^h)}}} + \\
&\qquad \frac{h_{len}}{2\,\|\mathbf{u}^h\|}\cdot\frac{d}{d\zeta}\bigg|_{\zeta=0}\frac{1}{\sqrt{1+\frac{9}{Re^2(\mathbf{u}^h)}}}
\end{aligned}
$$

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = \frac{h_{len}}{2}\cdot\left[-\frac{1}{2\,(\mathbf{u}^h\cdot\mathbf{u}^h)^{3/2}}\right]2\,\mathbf{u}^h\cdot\Delta\mathbf{u}^h\cdot\frac{1}{\sqrt{1+\frac{9}{Re^2(\mathbf{u}^h)}}}+$$

$$\frac{h_{len}}{2\|\mathbf{u}^h\|}\cdot\left[-\frac{1}{2\left(1+\frac{9}{Re^2(\mathbf{u}^h)}\right)^{3/2}}\right]\cdot\left(-\frac{2\cdot 9}{Re^3(\mathbf{u}^h)}\right)\cdot\frac{d}{d\zeta}\bigg|_{\zeta=0}Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)$$

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = -\frac{h_{len}}{2\|\mathbf{u}^h\|^3}\cdot\frac{\mathbf{u}^h\cdot\Delta\mathbf{u}^h}{\sqrt{1+\frac{9}{Re^2(\mathbf{u}^h)}}}+$$

$$+\frac{9\,h_{len}}{2\,\|\mathbf{u}^h\|\,\left(1+\frac{9}{Re^2(\mathbf{u}^h)}\right)^{3/2}Re^3(\mathbf{u}^h)}\cdot\frac{d}{d\zeta}\bigg|_{\zeta=0}Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)$$

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = -\frac{\tau}{\|\mathbf{u}^h\|^2}\,\mathbf{u}^h\cdot\Delta\mathbf{u}^h+$$

$$+\frac{9\,\tau}{\left(1+\frac{9}{Re^2(\mathbf{u}^h)}\right)Re^3(\mathbf{u}^h)}\cdot\frac{d}{d\zeta}\bigg|_{\zeta=0}Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)$$

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = \left[\frac{9\,\|\mathbf{u}^h\|^2}{(Re^2(\mathbf{u}^h)+9)\,Re(\mathbf{u}^h)}\cdot\right.$$

$$\left.\cdot\frac{d}{d\zeta}\bigg|_{\zeta=0}Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)-\mathbf{u}^h\cdot\Delta\mathbf{u}^h\right]\frac{\tau}{\|\mathbf{u}^h\|^2} \quad \text{(A.1)}$$

A.1 Newtonian Fluid $\mu(\mathbf{u}^h)=\mu$

$$\frac{d}{d\zeta}\bigg|_{\zeta=0}Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h) = \frac{d}{d\zeta}\bigg|_{\zeta=0}\frac{\|\mathbf{u}^h+\zeta\Delta\mathbf{u}^h\|\,h_{len}\,\rho}{2\,\mu} =$$

$$= \frac{h_{len}\,\rho}{2\,\mu}\,\frac{d}{d\zeta}\bigg|_{\zeta=0}\sqrt{\mathbf{u}^h\cdot\mathbf{u}^h+2\,h\,\mathbf{u}^h\cdot\Delta\mathbf{u}^h+h^2\,\Delta\mathbf{u}^h\cdot\Delta\mathbf{u}^h} =$$

$$= \frac{h_{len}\,\rho}{2\,\mu}\frac{1}{2\sqrt{\mathbf{u}^h\cdot\mathbf{u}^h}}\cdot 2\,\mathbf{u}^h\cdot\Delta\mathbf{u}^h =$$

$$= \frac{h_{len}\,\rho}{2\,\mu}\frac{\mathbf{u}^h}{\|\mathbf{u}^h\|}\cdot\Delta\mathbf{u}^h = \frac{Re(\mathbf{u}^h)}{\|\mathbf{u}^h\|^2}\mathbf{u}^h\cdot\Delta\mathbf{u}^h \quad \text{(A.2)}$$

When A.2 is substituted into A.1 the following is obtained:

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = \left[\frac{9\,\|\mathbf{u}^h\|^2}{(Re^2(\mathbf{u}^h)+9)\,Re(\mathbf{u}^h)}\cdot\frac{Re(\mathbf{u}^h)}{\|\mathbf{u}^h\|^2}\mathbf{u}^h\cdot\Delta\mathbf{u}^h - \mathbf{u}^h\cdot\Delta\mathbf{u}^h\right]\frac{\tau}{\|\mathbf{u}^h\|^2}$$

$$= \boxed{\left[\frac{9}{(Re^2(\mathbf{u}^h)+9)} - 1\right]\frac{\tau\,\mathbf{u}^h}{\|\mathbf{u}^h\|^2}\cdot\Delta\mathbf{u}^h}$$

$$= \tau'(\mathbf{u}^h)\cdot\Delta\mathbf{u}^h \quad \text{where} \quad \frac{\tau'(\mathbf{u}^h)}{\tau} = \left[\frac{9}{(Re^2(\mathbf{u}^h)+9)} - 1\right]\frac{\mathbf{u}^h}{\|\mathbf{u}^h\|^2}$$

A.2 Non-Newtonian Fluid

$$\left.\frac{d}{d\zeta}\right|_{\zeta=0} Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h) = \frac{\|\mathbf{u}^h+\zeta\Delta\mathbf{u}^h\|\,h_{len}\,\rho}{2\,\mu(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)} =$$

$$= \frac{h_{len}\,\rho}{2\,\mu(\mathbf{u}^h)}\frac{\mathbf{u}^h}{\|\mathbf{u}^h\|}\cdot\Delta\mathbf{u}^h + \frac{\|\mathbf{u}^h\|\,h_{len}\,\rho}{2\,\mu(\mathbf{u}^h)}\left.\frac{d}{d\zeta}\right|_{\zeta=0}\frac{1}{\mu(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)}$$

$$\left.\frac{d}{d\zeta}\right|_{\zeta=0}\frac{1}{\mu(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h)} = -\frac{1}{\mu^2(\mathbf{u}^h)}\cdot\left.\frac{d}{d\zeta}\right|_{\zeta=0}\mu(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h) =$$

$$= -\frac{1}{\mu^2(\mathbf{u}^h)}\cdot\mu'(\mathbf{u}^h)\frac{2}{\dot\gamma}\cdot\left[\varepsilon(\mathbf{u}^h):\varepsilon(\Delta\mathbf{u}^h)\right]$$

$$\left.\frac{d}{d\zeta}\right|_{\zeta=0} Re(\mathbf{u}^h+\zeta\Delta\mathbf{u}^h) = \frac{h_{len}\,\rho}{2\,\mu(\mathbf{u}^h)}\frac{\mathbf{u}^h}{\|\mathbf{u}^h\|}\cdot\Delta\mathbf{u}^h -$$

$$- \frac{\|\mathbf{u}^h\|\,h_{len}\,\rho\,\mu'(\mathbf{u}^h)}{\mu^2(\mathbf{u}^h)\,\dot\gamma}\cdot\left[\varepsilon(\mathbf{u}^h):\varepsilon(\Delta\mathbf{u}^h)\right]$$

$$= \frac{Re(\mathbf{u}^h)\,\mathbf{u}^h}{\|\mathbf{u}^h\|^2}\cdot\Delta\mathbf{u}^h - \frac{2\,Re(\mathbf{u}^h)\,\mu'(\mathbf{u}^h)}{\mu(\mathbf{u}^h)\,\dot\gamma}\cdot\left[\varepsilon(\mathbf{u}^h):\varepsilon(\Delta\mathbf{u}^h)\right] \qquad \text{(A.3)}$$

When A.3 is substituted into A.1 the following is obtained:

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = \left\{\frac{9\,\|\mathbf{u}^h\|^2}{(Re^2(\mathbf{u}^h)+9)\,Re(\mathbf{u}^h)}\cdot\right.$$

$$\left.\left[\frac{Re(\mathbf{u}^h)}{\|\mathbf{u}^h\|^2}\mathbf{u}^h\cdot\Delta\mathbf{u}^h - \frac{2\,Re(\mathbf{u}^h)\,\mu'(\mathbf{u}^h)}{\mu(\mathbf{u}^h)\,\dot\gamma}\cdot\left[\varepsilon(\mathbf{u}^h):\varepsilon(\Delta\mathbf{u}^h)\right]\right] - \mathbf{u}^h\cdot\Delta\mathbf{u}^h\right\}\frac{\tau}{\|\mathbf{u}^h\|^2} =$$

$$\boxed{\left[\frac{9}{(Re^2(\mathbf{u}^h)+9)} - 1\right]\frac{\tau\,\mathbf{u}^h}{\|\mathbf{u}^h\|^2}\cdot\Delta\mathbf{u}^h - \frac{18\,\tau\,\mu'(\mathbf{u}^h)}{(Re^2(\mathbf{u}^h)+9)\,\mu(\mathbf{u}^h)\,\dot\gamma}\cdot\left[\varepsilon(\mathbf{u}^h):\varepsilon(\Delta\mathbf{u}^h)\right]}$$

or if $\Delta\mathbf{u}^h = \mathrm{N_B}\Delta\mathbf{u_B}$

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] =$$
$$\left[\frac{9}{(Re^2(\mathbf{u})+9)} - 1\right] \frac{\tau\,\mathbf{u}}{\|\mathbf{u}\|^2}\cdot\mathrm{N_B}\Delta\mathbf{u_B} - \frac{18\,\tau\,\mu'(\mathbf{u})}{(Re^2(\mathbf{u})+9)\;\mu(\mathbf{u})\,\dot\gamma}\cdot[\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathrm{N_B}\Delta\mathbf{u_B})] =$$
$$\left[\frac{9}{(Re^2(\mathbf{u})+9)} - 1\right] \frac{\mathrm{N_B}\tau\,\mathbf{u}}{\|\mathbf{u}\|^2}\cdot\Delta\mathbf{u_B} - \frac{18\,\tau\,\mu'(\mathbf{u})}{(Re^2(\mathbf{u})+9)\;\mu(\mathbf{u})\,\dot\gamma}\,(\boldsymbol{\varepsilon}(\mathbf{u})\nabla\mathrm{N_B})\cdot\Delta\mathbf{u_B}$$

$$\mathcal{D}\tau(\mathbf{u}^h)[\Delta\mathbf{u}^h] = \boldsymbol{\tau}'(\mathbf{u})\cdot\Delta\mathbf{u_B}$$

where

$$\frac{\boldsymbol{\tau}'(\mathbf{u})}{\tau} = \left[\frac{9}{(Re^2(\mathbf{u})+9)} - 1\right] \frac{\mathrm{N_B}}{\|\mathbf{u}\|^2}\mathbf{u} - \frac{18\,\mu'(\mathbf{u})}{(Re^2(\mathbf{u})+9)\;\mu(\mathbf{u})\,\dot\gamma}\boldsymbol{\varepsilon}(\mathbf{u})\nabla\mathrm{N_B}$$

Appendix B

Common Identities

This appendix contains a list of identities that are used throughout thesis. The identities are included with their accompanying proofs. The identities are defined via an index notation. This means that repeated indexes imply summation. For instance:

$$a_i b_i = \begin{cases} \sum_{i=1}^{2} a_i b_i & \text{2D case} \\ \sum_{i=1}^{3} a_i b_i & \text{3D case} \end{cases}$$

The list of common identities:

- $\mathbf{a} \cdot \mathbf{b} = a_i b_i$
- $(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j$
- $(\mathbf{a} \otimes \mathbf{b})\mathbf{x} = (\mathbf{b} \cdot \mathbf{x})\mathbf{a}$
- $\mathbf{e}_i \otimes \mathbf{e}_j = \mathbf{I}$ where $(\mathbf{e}_1, \mathbf{e}_2)$ is the orthonormal basis of R^2
- $\nabla \cdot \mathbf{a} = a_{i,i}$
- $\nabla \mathbf{a} = a_{i,j}$
- $(\mathbf{b} \cdot \nabla)\mathbf{a} = b_j a_{i,j}$
- $\nabla(\alpha \mathbf{a}) = \mathbf{a} \otimes \nabla\alpha + \alpha \nabla \mathbf{a}$
  Proof:
  $\nabla(\alpha \mathbf{a}) = (\alpha a_i)_{,j}$
  $= \alpha_{,j} a_i + \alpha a_{i,j}$
  $= \mathbf{a} \otimes \nabla\alpha + \alpha \nabla \mathbf{a}$

- $\nabla \cdot (\alpha \mathbf{a}) = \nabla\alpha \cdot \mathbf{a} + \alpha(\nabla \cdot \mathbf{a})$
  Proof:

  $$\begin{aligned}\nabla \cdot (\alpha \mathbf{a}) &= (\alpha a_i)_{,i} \\ &= \alpha_{,i} a_i + \alpha a_{i,i} \\ &= \nabla\alpha \cdot \mathbf{a} + \alpha(\nabla \cdot \mathbf{a})\end{aligned}$$

- $\nabla \cdot (\mathbf{a} \otimes \mathbf{b}) = (\mathbf{b} \cdot \nabla)\mathbf{a} + (\nabla \cdot \mathbf{b})\mathbf{a}$
  Proof:

  $$\begin{aligned}\nabla \cdot (\mathbf{a} \otimes \mathbf{b}) &= (a_i b_j)_{,j} \\ &= a_{i,j} b_j + a_i b_{j,j} \\ &= (\mathbf{b} \cdot \nabla)\mathbf{a} + (\nabla \cdot \mathbf{b})\mathbf{a}\end{aligned}$$

- $\nabla \cdot (\boldsymbol{\sigma}\, \mathbf{u}) = (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{u} + \boldsymbol{\sigma} : \nabla^T \mathbf{u}$
  Proof:

  $$\begin{aligned}\nabla \cdot (\boldsymbol{\sigma}\, \mathbf{u}) &= (\sigma_{ij} u_j)_{,i} \\ &= \sigma_{ij,i} u_j + \sigma_{ij} u_{j,i} \\ &= (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{u} + \boldsymbol{\sigma} : \nabla^T \mathbf{u}\end{aligned}$$

- $(\mathbf{a} \otimes \mathbf{b}) : \mathbf{I} = \mathbf{a} \cdot \mathbf{b}$
  Proof:

  $$\begin{aligned}(\mathbf{a} \otimes \mathbf{b}) : \mathbf{I} &= a_i b_j \mathrm{I}_{ij} \\ &= a_i b_i \qquad \text{where } \mathrm{I}_{ij} = \begin{cases} 1 : i = j \\ 0 : i \neq j \end{cases} \\ &= \mathbf{a} \cdot \mathbf{b}\end{aligned}$$

Appendix C

Using symbolic computation

The role of this appendix is to elaborate on the process of converting the values of internal forces and stiffness matrices that were defined in chapter 5 to the finite element code. This appendix assumes that the finite element code is written in FORTRAN. This is the case with general purpose code Elfen which was used as an implementation platform. There in no special limitation for using the same procedure for a C/C++ finite element program. The only difference is that for a C/C++ finite element implementation certain steps can be dropped which are due to the limitations the FORTRAN programming language.

The procedure uses the Mathematica package to perform the basic algebraic manipulations and conversion of mathematical expressions into FORTRAN code (Wolfram, 1996). The conversion is based on use of the command **FortranForm**. This command takes an expression as an argument and returns the expression in FORTRAN form.

The automation is done in several steps. The first step is to rename the factors that make the expression into names that are more practical. Certain factors that are constant and do not change within the relevant loops can also be grouped into a single factor. For instance:

$$\begin{aligned} \tau &\longrightarrow TAU \\ \mathrm{N}_{\mathrm{B}}^{gp} &\longrightarrow SHPB \\ (\mathbf{u}_{n+\gamma}^{gp} \cdot \nabla \mathrm{N}_{\mathrm{A}}) &\longrightarrow UGNAA \\ \nabla \mathbf{u}_{n+\alpha} &\longrightarrow VELOGRM \end{aligned}$$

The next step is to define the types that every expression belongs to. For

instance

$$\begin{bmatrix} \text{typeUU} & \vdots & \text{typeUP} \\ \cdots & \cdots & \cdots \\ \text{typePU} & \vdots & \text{typePP} \end{bmatrix} \begin{bmatrix} \text{U} \\ \cdots \\ \text{P} \end{bmatrix} = \begin{bmatrix} \text{typeU} \\ \cdots \\ \text{typeP} \end{bmatrix} \tag{C.1}$$

Once the expressions are translated using the relevant definitions they can be entered into the mathematica program that generates the corresponding FORTRAN code. A flow chart that summarises this can be found in Figure C.1.

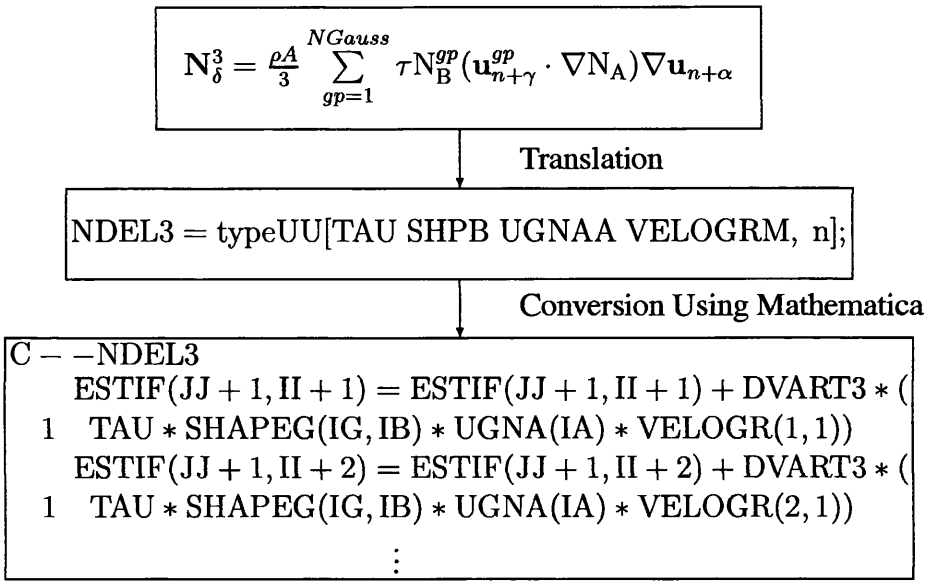


Figure C.1: From equations to code

The mathematica program that performs the conversion is organised in one notebook that calls all the relevant functions and creates the stiffness matrices and internal force vectors. At the beginning of the notebook the directories are set and the "ForceCalc" and "StiffCalc" package are read in. Then the force and stiff functions are called both for 2D and 3D versions. The dimension are set by setting n either to 2 or 3. After every call to force or stiff a batch file "cp.bat" is called that post processes the result into a form that can be incorporated directly into the code. The source file for "MakeAll" mathematica notebook is as follows:

```
Remove["Global`*"];

Off[General::spell];
```

```
Off[General::"spell1 "];

workDir = "K:
Projects
ConvergenceError
Version2";

force2DDir = workDir;

force3DDir = workDir;

stiff2DDir = workDir;

stiff3DDir = workDir;

SetDirectory[workDir];

Get["ForceCalc`"];

Get["StiffCalc`"];

force[2, force2DDir, "force2D"];
Run[force2DDir <> "
cp.bat", force2D, NP];

force[3, force3DDir, "force3D"];
Run[force2DDir <> "
cp.bat", force3D, NP];

stiff[2,stiff2DDir, "stiff2D"];
Run[stiff2DDir <> "
cp.bat", stiff2D, NP];

stiff[3, stiff3DDir, "stiff3D"];
Run[stiff3DDir <> "
cp.bat", stiff3D ,NP];
```

This is the source file for "ForceCalc" package:

```
BeginPackage["ForceCalc`", "OutputUtils`"]

force::usage =
     "force[n,dir] n is equal to
      2 for 2D and 3 for 3D, dir outpur directory"

Begin["OutputUtils`"]

force[n_,dir_,outf_]:= Module[
  {n1gp, n3gp},
  SetDirectory[dir];
```

```
(* Define variables used latter *)
n1gp = 4;
n3gp = 4;
ISTR = {"II+1", "II+2", "II+3", "II+4"};
ND1GP = {{"K", "DVAVIS"}, {"G", "DVAT3"},
         {"GTR", "DVAT3"}, {"GEPS", "DVATR"}};
ND3GP = {{"NN", "DVART3"}, {"NDEL", "DVART3"},
         {"NEPS", "DVAT3"}, {"GDEL", "DVAT3"}};
IDEN = 1.0 * IdentityMatrix[n];
DNDAL = {{DNDX[IA], DNDY[IA]}, {DNDX[IA], DNDY[IA], DNDZ[IA]}};
DNDBL = {{DNDX[IB], DNDY[IB]}, {DNDX[IB], DNDY[IB], DNDZ[IB]}};
STRNT2 = { {STRNG[1], 0.5STRNG[3]}, {0.5STRNG[3], STRNG[2]} };
STRNT3 = { {STRNG[1], 0.5STRNG[4], 0.5STRNG[5]},
           {0.5STRNG[4], STRNG[2], 0.5STRNG[6]},
           {0.5STRNG[5], 0.5STRNG[6], STRNG[3]} };
STRNTL = {   STRNT2, STRNT3   };
TAUPRML = {{TAUPRM[1, IB], TAUPRM[2, IB]},
           {TAUPRM[1, IB], TAUPRM[2, IB], TAUPRM[3, IB]}};
UBL = {{ELDISP[1, IB], ELDISP[2, IB]},
       {ELDISP[1, IB], ELDISP[2, IB], ELDISP[3, IB]}};
PBL = {ELDISP[3, IB], ELDISP[4, IB]};

SHPA = SHAPEG[IG, IA];
SHPB = SHAPEG[IG, IB];
UGNAA = UGNA[IA];
UGNBB = UGNB[IB];
DNDA = DNDAL[[n - 1]];
DNDB = DNDBL[[n - 1]];
UB = UBL[[n - 1]];
PB = PBL[[n - 1]];
TAUPRMV = TAUPRML[[n - 1]];
UJUIJV = Array[UJUIJ, n];
STRNT = STRNTL[[n - 1]];
VELOGRM = Array[VELOGR, {n, n}];
PRESGRV = Array[PRESGR, n];
GNAGPRV = DNDA . PRESGRV;
REPL1 = Table[0,{n1gp}];
REPL3 = Table[0,{n3gp}];

(* Beginning of definition of Force vector *)
GEPS = typeP[-TAU PB DNDA .DNDB, n];
K = typeU[ ExpandAll[1.0((UB ⊗ DNDB + DNDB ⊗ UB). DNDA)], n];
G = typeU[-DNDA PB, n];
GTR = typeP[-DNDB . UB, n];
NN = typeU[SHPA*UJUIJV, n];
NDEL = typeU[TAU UGNAA UJUIJV, n];
NEPS = typeP[-TAU UJUIJV. DNDA, n];
GDEL = typeU[TAU UGNAA PRESGRV, n];
(* End of definition of Force vector *)

(* Replace numbers with variables that are used in
   ELFEN i.e. 0.5 -> RP5 or 2.0 -> R2 *)
```

```
  Do[ REPL1[[I]] = replvector[ Simplify[ ToExpression[
    ND1GP[[I, 1]] ] ] ] /. x_^2 -> HoldForm[x x] , {I, n1gp} ];
  Do[ REPL3[[I]] = replvector[ Simplify[ ToExpression[
    ND3GP[[I, 1]] ] ] ] /. x_^2 -> HoldForm[x x] , {I, n3gp} ];

  (* Output values to file *)
  Do[outvectorcom[ "RLOAD", ISTR, ND3GP[[I, 2]], REPL3[[I]],
     outf <> ND3GP[[I, 1]] <> "G.F", "C--" <> ND3GP[[I, 1]] <>
     "\n"], {I, n3gp}];
  Do[outvectorcom[ "RLOAD", ISTR, ND1GP[[I, 2]], REPL1[[I]],
     outf <> ND1GP[[I, 1]] <> "S.F", "C--" <> ND1GP[[I, 1]] <>
     "\n"], {I, n1gp}];
]

End[]

EndPackage[]
```

This is the source file for "StiffCalc" package:

```
BeginPackage["StiffCalc`", "OutputUtils`"]

stiff::usage =
  "stiff[n,dir] n is equal to
   2 for 2D and 3 for 3D, dir outpur directory"

Begin["OutputUtils`"]

stiff[n_,dir_,outf_]:= Module[
  {n1gp,n3gp},
  SetDirectory[dir];

  (* Define variables used latter *)
  n1gp = 5;
  n3gp = 13;
  JSTR = {"JJ+1", "JJ+2", "JJ+3", "JJ+4"};
  ISTR = {"II+1", "II+2", "II+3", "II+4"};
  ND1GP = {{"K1", "DVAVIS"}, {"K2", "DVDVDS"},
           {"G", "DVAT3"}, {"GTR", "DVAT3"},
           {"GEPS2", "DVATR"}};
  ND3GP = {{"N1", "DVART3"}, {"N2", "DVART3"},
           {"NDEL1", "DVART3"}, {"NDEL2", "DVART9"},
           {"NDEL3", "DVART3"}, {"NDEL4", "DVART3"},
           {"NEPS1", "DVAT3"}, {"NEPS2", "DVAT3"},
           {"NEPS3", "DVAT3"}, {"GDEL1", "DVAT3"},
           {"GDEL2", "DVAT9"}, {"GDEL3", "DVAT3"},
           {"GEPS1", "DVAT3R"}};
  IDEN = 1.0*IdentityMatrix[n];
  DNDAL = {{DNDX[IA], DNDY[IA]}, {DNDX[IA], DNDY[IA], DNDZ[IA]}};
  DNDBL = {{DNDX[IB], DNDY[IB]}, {DNDX[IB], DNDY[IB], DNDZ[IB]}};
  STRNT2 = { {STRNG[1], 0.5STRNG[3]}, {0.5STRNG[3], STRNG[2]} };
```

```
STRNT3 = { {STRNG[1], 0.5STRNG[4], 0.5STRNG[5]},
           {0.5STRNG[4], STRNG[2], 0.5STRNG[6]},
           {0.5STRNG[5], 0.5STRNG[6], STRNG[3]} };
STRNTL = { STRNT2, STRNT3 };
TAUPRML = {{TAUPRM[1, IB], TAUPRM[2, IB]},
           {TAUPRM[1, IB], TAUPRM[2, IB], TAUPRM[3, IB]}};
UBL = {{ELDISP[1, IB], ELDISP[2, IB]},
       {ELDISP[1, IB], ELDISP[2, IB], ELDISP[3, IB]}};
PBL = {ELDISP[3, IB], ELDISP[4, IB]};

SHPA = SHAPEG[IG, IA];
SHPB = SHAPEG[IG, IB];
UGNAA = UGNA[IA];
UGNBB = UGNB[IB];
DNDA = DNDAL[[n - 1]];
DNDB = DNDBL[[n - 1]];
PB = PBL[[n - 1]];
TAUPRMV = TAUPRML[[n - 1]];
UJUIJV = Array[UJUIJ, n];
STRNT = STRNTL[[n - 1]];
VELOGRM = Array[VELOGR, {n, n}];
PRESGRV = Array[PRESGR, n];
GNAGPRV = DNDA . PRESGRV;
REPL1 = Table[0,{n1gp}];
REPL3 = Table[0,{n3gp}];

(* Begining of definition of Stiffness matrix *)
K1 = typeUU[(DNDA .DNDB) IDEN + (DNDB ⊗ DNDA), n];
K2 = typeUU[(STRNT .  DNDA) ⊗ (STRNT . DNDB), n];
G = typeUP[-DNDA, n];
GTR = typePU[-DNDB, n];
GEPS2 = typePP[-TAU (DNDA . DNDB), n];
N1 = typeUU[SHPA*SHPB*VELOGRM, n];
N2 = typeUU[SHPA*UGNBB*IDEN, n];
NDEL1 = typeUU[TAU UGNAA (UJUIJV ⊗ TAUPRMV), n];
NDEL2 = typeUU[TAU (UJUIJV ⊗ DNDA), n];
NDEL3 = typeUU[TAU SHPB UGNAA VELOGRM, n];
NDEL4 = typeUU[TAU UGNAA UGNBB IDEN, n];
NEPS1 = typePU[-TAU (TAUPRMV ⊗ UJUIJV). DNDA, n];
NEPS2 = typePU[-TAU SHPB Transpose[VELOGRM]. DNDA, n];
NEPS3 = typePU[-TAU UGNBB DNDA, n];
GDEL1 = typeUU[TAU UGNAA (PRESGRV ⊗ TAUPRMV), n];
GDEL2 = typeUU[TAU (PRESGRV ⊗ DNDA), n];
GDEL3 = typeUP[TAU UGNAA DNDB, n ];
GEPS1 = typePU[-TAU GNAGPRV  TAUPRMV, n];
(* End of definition of Stiffness matrix *)

(* Replace numbers with variables that are used in
   ELFEN i.e. 0.5 -> RP5 or 2.0 -> R2 *)
Do[ REPL1[[I]] = replarray[ Simplify[ ToExpression[
  ND1GP[[I, 1]] ] ] ] /. x_^2 -> HoldForm[x x] , {I, n1gp} ];
Do[ REPL3[[I]] = replarray[ Simplify[ ToExpression[
```

```
    ND3GP[[I, 1]] ] ] ] /. x_^2 -> HoldForm[x x] , {I, n3gp} ];

  (* Output values to file *)
  Do[outarraycom[ "ESTIF", JSTR, ISTR, ND1GP[[I, 2]],
     Transpose[ REPL1[[I]] ], outf <> ND1GP[[I, 1]] <>
     "S.F", "C--" <> ND1GP[[I, 1]] <> "\n"], {I, n1gp}];
  Do[outarraycom[ "ESTIF", JSTR, ISTR, ND3GP[[I, 2]],
     Transpose[ REPL3[[I]] ], outf <> ND3GP[[I, 1]] <>
     "G.F", "C--" <> ND3GP[[I, 1]] <> "\n"], {I, n3gp}];
]

End[]

EndPackage[]
```

Apart from the "ForceCalc" and "StiffCalc" packages there is also an additional package that contains all the output functions and is called "OutputUtils" package. It is automatically called from "ForceCalc" and "StiffCalc".
The source file for "OutputUtils" package is as follows:

```
BeginPackage["OutputUtils`"]

tr::usage =
     "tr[a_]"

makesublist::usage =
     "makesublist[kList_, grad_, i_, corr_]"

makePosLists::usage =
     "makePosLists[kList_, gradni_, gradnj_]"

inner::usage =
     "inner[x_, y_]"

outarraycom::usage =
     "outarraycom[matstr_, iistr_, jjstr_,
                  dvolustr_, mat_, filename_,comstr_]"

outvectorcom::usage =
     "outvectorcom[vecstr_, iistr_, dvolustr_,
                   vec_, filename_,comstr_]"

replarray::usage =
     "replarray[mat_]"

replvector::usage =
     "replvector[vec_]"

typeUU::usage =
     "typeUU[mat_,n_]"
```

```
typeUP::usage =
     "typeUP[mat_,n_]"

typePU::usage =
     "typePU[mat_,n_]"

typePP::usage =
     "typePP[mat_,n_]"

typeU::usage =
     "typeU[vec_,n_]"

typeP::usage =
     "typeP[vec_,n_]"

CircleTimes::usage =
     "(x_)  ⊗  (y_)"

Begin["`Private`"]

(x_) ⊗ (y_) := Outer[Times, x, y];

inner[x_, y_] := tr[Transpose[x] . y]

tr[a_] := Module[{n, diag, dims, i},
   dims = Dimensions[a]; n = dims[[1]];
   diag = Table[a[[i,i]], {i, n}];
   diag = Flatten[diag]; Plus @@ diag]

makesublist[kList_, grad_, i_, corr_] := Module[
  {temp, k},
  temp = Position[kList, grad[[i,corr]]];
  Table[temp[[k,1]], {k, Dimensions[temp][[1]]}]
]

makePosLists[kList_, gradni_, gradnj_] := Module[
   {i, corr, rdim, posarrayi, posarrayj},
   rdim = Dimensions[gradni][[1]];
   posarrayi =
     Table[Join @@ Table[makesublist[kList, gradni, i, corr],
      {corr, rdim}], {i, rdim}];
   posarrayj =
     Table[Join @@ Table[makesublist[kList, gradnj, i, corr],
      {corr, rdim}],{i, rdim}];
   {posarrayi, posarrayj}
]

outarraycom[matstr_, iistr_, jjstr_,
            dvolustr_, mat_, filename_,comstr_] := Module[
  {i, j, m, n, file, str1, str2, tab, tab1},
  file = OpenWrite[filename];
  m = Dimensions[mat][[1]];
```

```
  n = Dimensions[mat][[2]];
  tab = "        ";
  tab1 = StringJoin["     1", tab];
  WriteString[file,comstr];
  Do[If[mat[[i,j]]==0,Continue[]];
    str1 = StringJoin[matstr, "(", iistr[[i]],
                      ",", jjstr[[j]], ")"];
    str2 = StringJoin[" = ", str1, "+", dvolustr, "*(\n"];
    (WriteString[file, tab, str1, str2, tab1,
    FortranForm[mat[[i,j]]], " )\n"]; ),
    {i, 1, m}, {j, 1, n}];
  Close[file];
]

outvectorcom[vecstr_, iistr_,
             dvolustr_, vec_, filename_,comstr_] := Module[
  {i, n, file, str1, str2, tab, tab1},
  file = OpenWrite[filename];
  n = Dimensions[vec][[1]];
  tab = "        ";
  tab1 = StringJoin["     1", tab];
  WriteString[file,comstr];
  Do[str1 = StringJoin[vecstr, "(", iistr[[i]], ")"];
     str2 = StringJoin[" = ", str1, "+", dvolustr, "*(\n"];
     If[vec[[i]]==0,Continue[]];
     (WriteString[file, tab, str1, str2, tab1,
     FortranForm[vec[[i]]], " )\n"]; ),
     {i, 1, n}];
  Close[file];
]

replarray[mat_] := Module[
  {i,j,m,n,outmat},
  m = Dimensions[mat][[1]];
  n = Dimensions[mat][[2]];
  outmat = Table[ToExpression[
    ToUpperCase[
      StringReplace[
        ToString[mat[[i,j]]],{"2."->"R2","1."->"","0.5"->"RP5",
            "0.25"->"RP25","0.125"->"RP125"}]]],{i,m},{j,n}];
  outmat
]

replvector[vec_] := Module[
 {i,n,outvec},
 n = Dimensions[vec][[1]];
 outvec = Table[ToExpression[
   ToUpperCase[
     StringReplace[
       ToString[vec[[i]]],{"2."->"R2","1."->"","0.5"->"RP5",
         "0.25"->"RP25","0.125"->"RP125"}]]],{i,n}];
 outvec
```

```
]

typeUU[mat_,n_] := Module[
  {i,j,outmat},
  outmat=Table[0,{i,n+1},{j,n+1}];
  Do[outmat[[i,j]]=mat[[i,j]],{i,n},{j,n}];
  outmat
]

typeUP[vec_,n_] := Module[
  {i,j,outmat},
  outmat=Table[0,{i,n+1},{j,n+1}];
  Do[outmat[[i,n+1]]=vec[[i]],{i,n}];
  outmat
]

typePU[vec_,n_] := Module[
  {i,j,outmat},
  outmat=Table[0,{i,n+1},{j,n+1}];
  Do[outmat[[n+1,j]]=vec[[j]],{j,n}];
  outmat
]

typePP[scal_,n_] := Module[
  {i,j,outmat},
  outmat=Table[0,{i,n+1},{j,n+1}];
  outmat[[n+1,n+1]]=scal;
  outmat
]

typeU[vec_,n_] := Module[
  {i,outvec},
  outvec=Table[0,{i,n+1}];
  Do[outvec[[i]]=vec[[i]],{i,n}];
  outvec
]

typeP[scal_,n_] := Module[
  {i,outvec},
  outvec=Table[0,{i,n+1}];
  outvec[[n+1]]=scal;
  outvec
]

End[]

EndPackage[]
```

This is the contents of the "cp.bat" batch file:

```
rem Delete old values
del *.fix
```

```
del %1fs.f
del %1fg.f

rem Wrap Fortran files so they don't exceed 72 characters
rem don't start wrapping before line 55
for %%1 in (%1*.f) do elfenwrap %%1 55 72

rem Merge s and g files                                          10
copy %1*g.f.fix /b %1fg.f.keep /b
copy %1*s.f.fix /b %1fs.f.keep /b

rem Remove temporary files
del %1*.fix
del %1*.f
move %1fg.f.keep %1fg.f
move %1fs.f.keep %1fs.f

rem Pause if requested                                           20
if %2 == P pause
```

The purpose of this file is to post process the FORTRAN files. It has two purposes. The first is to wrap the FORTRAN file so they conform to the FORTRAN limitation for line lengths. To do this a program "wrap" is used. It takes three parameters, the name of the file to wrap, the position of the possible breaking points and the maximum length of the line that is allowed. For FORTRAN files this length is equal to 72. For C/C++ there is no limitations so this step can be skipped.

The "wrap" program consists of the main file "wrap.c" and the file "util.c" that contains the necessary subroutines.
This is the content of the "wrap.c" file.

```
/*************************************************************
 *                                                           *
 *   A program to wrap fortran file to conform to the        *
 *   correct line length of 72 characters per line           *
 *                                                           *
 *   Usage:                                                  *
 *   wrap <FORTRAN File> <Start Breaking> <Maximum length>   *
 *   Initial coding: S.Slijepcevic                           *
 *                                                           *
 *************************************************************/                10

#include <stdio.h>
#include <io.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
#include "util.h"

int main( int argc, char *argv[] )
{                                                                20
```

```
FILE *infile, *outfile;
char infilename[100],outfilename[100];
int  maxline, minline;
int  filehandle, maxlinesize;
long filesize;
char *linestr, *substring, *outstring, tab[73];
int  substrsize, linestrsize, tabsize, subbp;
int  mintextsize, maxtextsize, textskip;
char counter, *orglinestr,*orgsubstring, *orgoutstring;                    30


if ( argc > 3 ) {
  sprintf(infilename, "%s", argv[1] );
  sscanf(argv[2] , "%d", &minline );
  sscanf(argv[3] , "%d", &maxline );
} else {
  printf( "USAGE : elfenwrap <filename> minline maxline\n\n" );
  return 1;
};                                                                         40

sprintf( outfilename, "%s.fix", infilename );

if (   ( (infile=fopen(infilename,"r"))==NULL )     ) {
  printf( "File %s does not exist \n\n", infilename );
  return 1;
}
if (   ( (outfile=fopen(outfilename,"w"))==NULL )     ) {
  printf( "Error opening file %s \n\n", outfilename );
  return 1;                                                                50
}
else {
  filehandle=_fileno(infile);
  filesize=_filelength(filehandle);
  maxlinesize=filesize+1;
  printf( "Opened file: %s size: %i max Line: %i\n",infilename,filesize,maxlinesize );
  printf( "Fixed  file: %s\n",outfilename );
  rewind( infile );
  rewind( outfile );
  linestr = (char *) malloc(maxlinesize);                                  60
  substring = (char *) malloc(maxline-minline+1);
  outstring = (char *) malloc(maxline+1);
  if(   !( linestr ) || !( substring ) || !( outstring ) ) {
     printf( "Error allocating memory!!!\n" );
    return 1;
  };
  orglinestr    = linestr;
  orgsubstring = substring;
  orgoutstring = outstring;
  do {                                                                     70
    linestr    = orglinestr;
    substring = orgsubstring;
```

```
        outstring = orgoutstring;
        linestr[0]      = 0;
        substring[0] = 0;
        outstring[0] = 0;
        skipline2(infile,outfile,linestr,maxlinesize);
/*        printf("%s\n",linestr);*/
        if(    linestr[5]=='1'    && (int) strlen(linestr) > maxline        ) {
          tabsize=gettab(linestr,tab);                                          80
          textskip=tabsize+6;
          maxtextsize=maxline-textskip;
          mintextsize=minline-textskip;
          counter=linestr[5];
          linestr=&linestr[tabsize+6];
          substrsize= maxline - minline;
          while( (linestrsize=(int) strlen(linestr)) > maxtextsize ) {
            outstring[0]=0;
            strncpy(substring, &linestr[mintextsize],substrsize);
            substring[substrsize]=0;                                            90
            subbp=getbreakpoint(substring);
            sprintf(outstring,"          %c%s",counter,tab);
            strncat(outstring, linestr, mintextsize+subbp);
            fprintf(outfile,"%s\n",outstring);
            linestr=&linestr[mintextsize+subbp];
            counter++;
            if(counter==':') counter='0';
            if(counter=='P') counter='1';
          };
          sprintf(outstring,"          %c%s%s",counter,tab,linestr);           100
          fprintf(outfile,"%s",outstring);
        } else {
          fprintf(outfile,"%s",linestr);
        };
      } while ( !( feof(infile) ) );

      free(orgoutstring);
      free(orgsubstring);
      free(orglinestr);
      fclose(infile);                                                           110
      fclose(outfile);
      return 0;
    };
    return -1;
}
```

This is the content of the "util.h" file.

```
#include <stdio.h>
#include <string.h>

void skipline(FILE *infile, char* line, int linesize);
void skipline2(FILE *infile, FILE *outfile, char* line, int linesize);
void skipNlines(FILE *infile, char* line, int linesize, int n);
```

```
int gettab(char *linestr, char* tab);
int getbreakpoint(char* string);
```

This is the contents of the "util.c" file.

```
/*************************************************************
 *                                                           *
 *   Functions to manipulate strings and help read files     *
 *   Initial coding: S.Slijepcevic                           *
 *                                                           *
 *   List of functions:                                      *
 *   skipline      - get a new line while skiping comments   *
 *   skipline2     - get a new lines but output comments     *
 *   skipNlines    - get a new line but skip N lines         *
 *   gettab        - get the tab at the begining of a line   *
 *   getbreakpoint - get a break point                       *
 *                                                           *
 *************************************************************/

#include "util.h"

void skipline(FILE *infile, char* line, int linesize)
{
  int flag=1;
  while(flag)    {
    fgets(line,linesize,infile);
    if( line[0]==' ' || feof(infile) ) flag=0;
  };
}

void skipline2(FILE *infile, FILE *outfile, char* line, int linesize)
{
  int flag=1;
  while(flag)    {
    fgets(line,linesize,infile);
    if( line[0]==' ' || feof(infile) )
      flag=0;
    else {
      if( line[0]=='C' || line[0]=='c' )
        fprintf(outfile,"%s",line);
    };
  };
}

void skipNlines(FILE *infile, char* line, int linesize, int n)
{
  int i;
  for(i=0;i<n;i++) {
    fgets(line,linesize,infile);
  };
}
```

```
int gettab(char *linestr, char* tab)
{
  int tabstart=6, i;                                          50
  tab[0]=0;
  i=0;
  while( linestr[i+tabstart]==' ' ) {
    tab[i]=' ';
    tab[i+1]=0;
    i++;
  };
  return(i);
}
                                                              60
int getbreakpoint(char* string)
{
  int strsize, i,charnotf;
  strsize=strlen(string);
  i=strsize-1;
  charnotf=1;
  while ( i > -1 && charnotf) {
    switch(string[i]) {
    case '+':
      charnotf=0;                                             70
      break;
    case '-':
      charnotf=0;
      break;
    case '/':
      charnotf=0;
      break;
    case '*':
      if( i >0 ) {
        if( string[i-1]!='*' ) charnotf=0;                    80
      } else {
          i--;
      };
      break;
    default:
      i--;
    };
  };
  return(i+1);
}                                                             90
```

Appendix D

Computer Implementation

This appendix deals with the implementation of the non-Newtonian material models into the general purpose finite element program "Elfen". An overview of the routines that are used can be found in Table D.1. It is important to note that the same material routines are used for 2D, 3D, steady state and transient routines. This means that if a new material is added it is immediately available to all routines.

	2D		3D	
	Steady state	Transient	Steady state	Transient
Internal Force	pstfm.f	pstfmt.f	pstfso.f	pstfst.f
Stiffness Matrix	kfmb.f	kfmbt.f	kfsot.f	kfsott.f
Shape Functions	shp2dn.f		shp3dn.f	
Read Mat. Property	mtpfl1.f			
Calc. Mat. Parameters	mtfpst.f			

Table D.1: An overview of routines used

Bibliography

ALEXANDROU, A.N., MCGILVREAY, T.M., & BURGOS, G. 2001. Steady Herschel–Bulkley fluid flow in three–dimensional expansions. *Journal of Non-Newtonian Fluid Mechanics*, **100**, 77–96.

BABUŠKA, I. 1973. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, **20**, 179–192.

BALOCH, A., TOWNSEND, P., & WEBSTER, M. F. 1995. On two– and three–dimentional expansion flows. *Computers & Fluids*, **24**, 863–882.

BARNES, H.A. 1999. The yield stress – a review or '$\pi\alpha\nu\tau\alpha\ \rho\varepsilon\iota$' – everything flows? *Journal of Non-Newtonian Fluid Mechanics*, **81**, 133–178.

BARNES, H.A., & WALTERS, K. 1985. The yield stress myth? *Rheol. Acta*, **24**, 323–326.

BARNES, H.A., HUTTON, J.F., & WALTERS, K. 1989. *An Introduction to Rheology*. Amsterdam: Elsevier.

BEAULNE, M., & MITSOULIS, E. 1997. Creeping motion of a sphere in tubes filled with Herschel-Bulkley fluids. *Journal of Non-Newtonian Fluid Mechanics*, **72**, 55–71.

BEHR, M. 1992. *Stabilized Finite Element Methods for Incompressible Flows with Emphasis on Moving Boundaries and Interfaces.* Ph.D. thesis, University of Minnesota.

BEHR, M., & TEZDUYAR, T.E. 1994. Finite element solution strategies for large-scale flow simulations. *Computer Methods in Applied Mechanics and Engineering*, **112**, 3–24.

BERIS, A.N., TSAMOPOULOS, J.A., ARMSTRONG, R.C., & BROWN, R.A. 1985. Creeping motion of a sphere through a Bingham plastic. *Journal of Fluid Mechanics*, **158**, 219–244.

BEVERLY, C.R., & TANNER, R.I. 1992. Numerical analysis of three-dimensional Bingham plastic flow. *Journal of Non-Newtonian Fluid Mechanics*, **42**, 85–115.

BIRD, R. B., ARMSTRONG, R. C., & HASSAGER, O. 1977. *Dynamics of Polymeric Liquids - Fluid Mechanics.* New York: John Wiley.

BIRD, R.B., STEWART, W.E., & LIGHTFOOT, E.N. 2001. *Transport Phenomena.* Chichester: John Wiley.

BLACKERY, J., & MITSOULIS, E. 1997. Creeping motion of a sphere in tubes filled with a Bingham plastic material. *Journal of Non-Newtonian Fluid Mechanics,* **70**, 59–77.

BONET, J., & WOOD, R.D. 1997. *Nonlinear Continuum Mechanics for Finite Element Analysis.* Cambridge: Cambridge University Press.

BREZZI, F. 1974. On the existance, uniqueness and approximation of the saddle-point problems arising from Lagrangian multipliers. *Revue française d'automatique et de recherche opérationnelle,* **8**, 129–151.

BREZZI, F., & BATHE, K.J. 1990. A discourse on the stability conditions for mixed finite element formulations. *Computer Methods in Applied Mechanics and Engineering,* **82**, 27–57.

BREZZI, F., & FORTIN, M. 1991. *Mixed and Hybrid Finite Element Methods.* New York: Springer-Verlag.

BROOKS, A.N., & HUGHES, T.J.R. 1982. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes Equations. *Computer Methods in Applied Mechanics and Engineering,* **32**, 199–259.

CHHABRA, R.P. 1993. *Bubbles, Drops and Particles in Non-Newtonian Fluids.* Florida: CRC Press.

CHORIN, A.J., & MARSDEN, J.E. 1992. *A Mathematical Introduction to Fluid Mechanics.* Berlin: Springer.

CHUNG, T.J. 1978. *Finite Element Analysis in Fluid Dynamics.* New York: McGraw-Hill.

CIARLET, P.G. 1978. *Finite Element Methods for Elliptic Problems.* Amsterdam: North-Holland.

CROCHET, M.J., DAVIES, A.R., & WALTERS, K. 1984. *Numerical Simulation of Non-Newtonian Flow.* Amsterdam: Elsevier.

CURRIE, I.G. 1993. *Fundamental Mechanics of Fluids.* New York: McGraw-Hill.

DAVIS, P.J. 1963. *Interpolation and Approximation.* New York: Blaisdel.

DONEA, J. 1983. Recent advances in computational methods for steady and transient transport problems. *In: SMiRT-7.*

FLUMERFELT, R.W., PIERICK, M.W., COOPER, S.L., & BIRD, R.B. 1969. Generalized plane couette flow of a non-Newtonian fluid. *I & EC Fundamentals*, **8**(2), 354–357.

FRANCA, L.P. 2001. Advances in stabilized methods in computational mechanics. *Special Issue of Computer Methods in Applied Mechanics and Engineering*, **18**(1–2), 577–591.

GIRAULT, V., & RAVIART, P.A. 1979. *Finite Element Approximation of the Navier-Stokes Equations.* Berlin: Springer.

GRESHO, P. M., & SANI, R. L. 1998. *Incompressible Flow and the Finite Element Method, Advection-Diffusion and Isothermal Laminar Flow.* Chichester: John Wiley.

GROOVER, M.P. 1996. *Fundamentals of Modern Manufacturing: Materials, Processes and Systems.* New York: Prentice-Hall.

GUNASEKERA, J.S. 1989. *CAD/CAM of Dies.* Chichester: Ellis Horwood limited.

HARRIS, J. 1977. *Rheology and Non-Newtonian Flow.* New York: Longman.

HU, H., JOSEPH, D., & CROCHET, M.J. 1992. Direct Simulation of Fluid Particle Motions. *Theoretical and Computational Fluid Dynamics*, **3**, 285–306.

HUGHES, T.J.R. 1979. *Finite Element Methods for Convection Dominated Flows.* Vol. 34. New York: American Society of Mechanical Engineers.

HUGHES, T.J.R., & FRANCA, L.P. 1987. A new Finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces. *Computer Methods in Applied Mechanics and Engineering*, **65**, 85–96.

HUGHES, T.J.R., & MALLET, M. 1986a. A new Finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, **58**, 305–328.

HUGHES, T.J.R., & MALLET, M. 1986b. A new Finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, **58**, 329–336.

HUGHES, T.J.R., MALLET, M., & MIZUKAMI, A. 1986b. A new Finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering*, **54**, 341–355.

HUGHES, T.J.R., FRANCA, L.P., & BALESTRA, M. 1986a. A new Finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi Condition: A Stable Petrov-Galerkin Formulation of the Stokes Problem Accommodating Equal-Order Interpolations. *Computer Methods in Applied Mechanics and Engineering*, **59**, 85–99.

HUGHES, T.J.R., FRANCA, L.P., & MALLET, M. 1987. A new Finite element formulation for computational fluid dynamics: VI. Convergence analysis of the generalized SUPG formulation for linear time-dependent multidimensional advective-diffusive systems. *Computer Methods in Applied Mechanics and Engineering*, **63**, 97–112.

HUGHES, T.J.R., FRANCA, L.P., & HULBERT, G.M. 1989. A new Finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, **73**, 173–189.

JANSEN, K.E., WHITING, C.H., & HULBERT, G.M. 2000. A Generalized α–Method for Integrating the Filtered Navier–Stokes Equations with a Stabilized Finite Method. *Computer Methods in Applied Mechanics and Engineering*, **190**, 305–319.

JOHNSON, A.A., & TEZDUYAR, T.E. 1996a. 3D simulation of fluid-particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering*.

JOHNSON, A.A., & TEZDUYAR, T.E. 1996b. Parallel computation of incompressible flows with complex geometries. *International Journal for Numerical Methods in Fluids*.

JOHNSON, C. 1995. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge: Cambridge University Press.

JOHNSON, C., NÄVERT, U., & PITKÄRANTA, J. 1984. Finite element methods for linear hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering*, **45**, 285–312.

KARLO, V., & TEZDUYAR, T. 1998. 3D computation of unsteady flow past a sphere with parallel finite element method. *Computer Methods in Applied Mechanics and Engineering*, **151**, 267–276.

KREISS, H.O., & LORENZ, J. 1989. *Initial-Boundary Value Problems and the Navier-Stokes Equations.* Boston: Academic Press.

LAWAL, A., RAILKAR, S., & KALYON, D.M. 1996. Coupled Simulation of Die and Extrusion Flows Using Three-Dimensional Finite Element Method. *Society of Plastics Engineers ANTEC Technical Papers*, **42**, 155–159.

LIPSCOMB, G.G., & DENN, M.M. 1984. Flow of Bingham fluids in complex geometries. *Journal of Non-Newtonian Fluid Mechanics*, **14**, 337–346.

LIU, B.T., MULLER, S.J., & DENN, M.M. 2002. Convergence of a regularization method for creeping flow of a Bingham material about a rigid sphere. *Journal of Non-Newtonian Fluid Mechanics*, **102**, 179–191.

LUBLINER, J. 1990. *Plasticity Theory*. New York: Macmillan Publishing Company.

MICHAELI, W. 1992. *Extrusion Dies for Plastics and Rubber*. Munich: Hanser Publishers.

MITTAL, S., & TEZDUYAR, T.E. 1994. Massively Parallel finite element computation of incompressible flows involving fluid-body interaction. *Computer Methods in Applied Mechanics and Engineering*, **112**, 253–282.

O'BRIEN, K.T. 1992. *Computer Modeling for Extrusion and Other Continuous Polymer Processes – Applications*. Munich: Hanser Publishers.

O'DONOVAN, E.J., & TANNER, R.I. 1984. Numerical study of the Bingham squeeze film problem. *Journal of Non-Newtonian Fluid Mechanics*, **15**, 75–83.

ORTEGA, J.M., & RHEINBOLDT, W.C. 2000. *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Society for Industrial and Applied Mathematics (SIAM).

PAPANASTASIOU, T.C. 1987. Flow of materials with yield. *J. Rheol.*, **31**, 385–404.

PASHIAS, N., BOGER, D.V., SUMMERS, J., & GLENISTER, D.J. 1996. A Fifty Cent Rheometer for Yield Stress Measurement. *J. Rheol.*, **40**(6), 1179–1189.

PERIĆ, Đ. 1996. *Lecture notes: Continuum Mechanics Course*. Swansea.

PERIĆ, Đ., & SLIJEPČEVIĆ, S. 2001. Computational modeling of viscoplastic fluids based on a stabilised finite element method. *Engineering Computations*, **18**(3–4), 577–591.

PEYRET, R. 1996. *Computational Fluid Mechanics*. London: Academic Press.

REDDY, B.D. 1998. *Introductory Functional Analysis - With Application to Boundary Value Problems and Finite Elements*. Berlin: Springer.

SASTROHARTONO, T., & KWON, T.H. 1990. Finite element analysis of the mixing phenomena in tangential twin-screw extruders for non-newtonian fluids. *International Journal for Numerical Methods in Engineering*, **30**, 1369–1383.

SHAKIB, F., & HUGHES, T.J.R. 1991. A new Finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms. *Computer Methods in Applied Mechanics and Engineering*, **87**, 35–58.

SHAKIB, F., HUGHES, T.J.R., & JOHAN, Z. 1991. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, **89**, 141–219.

SIMO, J.C., & ARMERO, F. 1994. Unconditional stability and long-term behavior of transient algorithms for the incompressible Navier-Stokes and Euler equations. *Computer Methods in Applied Mechanics and Engineering*, **111**, 111–154.

SLIJEPČEVIĆ, S., & PERIĆ, Đ. 2000. Some aspects of computational modelling of non-Newtonian fluids based on stabilised finite element methods. *In: ECCOMAS 2000.*

SMYRNAIOS, D.N., & TSAMOPOULOS, J.A. 2001. Squeeze flow of Bingham plastics. *Journal of Non-Newtonian Fluid Mechanics*, **100**, 165–190.

SPURK, J.H. 1997. *Fluid Mechanics.* Berlin: Springer.

STOER, J., & BULIRSCH, R. 1980. *Introduction to Numerical Analysis.* New York: Springer.

STRANG, G., & FIX, G.J. 1973. *An Analysis of the Finite Element Method.* New York: Prentice-Hall.

TANNER, R.I. 1992. *Engineering Rheology.* Oxford: Oxford Science Publications.

TANNER, R.I., NICKELL, R.E., & BILGER, R.W. 1975. Finite Element Method for the Solution of Some Imcompressible non-Newtonian Fluid Mechanics Problems with Free Surfaces. *Computer Methods in Applied Mechanics and Engineering*, **6**, 155.

TEMAM, R. 1977. *Navier-Stokes equations.* Amsterdam: North-Holland.

TEZDUYAR, T.E., MITTAL, S., & R.SHIH. 1991. Time-accurate incompressible flow computations with quadrilateral velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, **87**, 363–384.

TEZDUYAR, T.E., MITTAL, S., RAY, S.E., & SHIH, R. 1992. Incompressible flow computations stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, **95**, 221–242.

TEZDUYAR, T.E., BEHR, M., & LIOU, J. 1994a. A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space-time procedure: I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*, **94**, 339–351.

TEZDUYAR, T.E., BEHR, M., & LIOU, J. 1994b. A new strategy for finite element computations involving moving boundaries and interfaces – The deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders. *Computer Methods in Applied Mechanics and Engineering*, **94**, 353–371.

THOMAS, D.G. 1961. Laminar-flow properties of flocculated suspensions. *A.I.Ch.E. Journal*, **7**, 431–437.

TUCKER, C.L. 1989. *Computer Modeling of Polymer Processing – Fundamentals.* Munich: Hanser Publishers.

WOLFRAM, S. 1996. *The Mathematica Book.* Cambridge: Cambridge University Press.

YANG, F. 1998. Exact solution for compressive flow of viscoplastic fluids under perfect slip wall boundary conditions. *Rheol. Acta*, **37**, 68–72.

ZAUDERER, E. 1989. *Partial Differential Equations of Applied Mathematics.* Chichester: John Wiley.

ZIENKIEWICZ, O.C, & TAYLOR, R.L. 2000. *The Finite Element Method.* Oxford: Butterworth-Heinemann.