Thermal Decomposition in 1D Shear Flow of Generalised Newtonian Fluids

Adeleke Olusegun Bankole (adeleke@aims.ac.za)
African Institute for Mathematical Sciences (AIMS)

Supervised by: Doctor Tirivanhu Chinyoka
University of Cape Town, South Africa.

22 May 2009

Submitted in partial fulfillment of a postgraduate diploma at AIMS

Abstract

The effect of variable viscosity in a thermally decomposable generalised Newtonian fluids (GNF) subjected to unsteady 1D shear flow is investigated by direct numerical simulations. A numerical algorithm based on the semi-implicit finite-difference scheme is implemented in time and space using the temperature corrected Carreau model equation as the model for variable viscosity. The chemical kinetics is assumed to follow the Arrhenius rate equation; this reaction is assumed to be exothermic. Approximate (Numerical) and graphical results are presented for some parameters in the problem, thermal criticality conditions are also discussed. It is observed that increasing the non-Newtonian nature of the fluid helps to delay the onset of thermal runaway when compared to Newtonian nature of the fluid.

Keywords: Thermally decomposable; Unsteady flow; Variable viscosity; Arrhenius chemical kinetics; Semi-implicit finite-difference scheme; Direct numerical simulations; Carreau model; Thermal runaway; Generalised Newtonian fluids(GNF)

Declaration

I, the undersigned, hereby declare that the work contained in this essay is my original work, and that any work done by others or by myself previously has been acknowledged and referenced accordingly.

Adeleke Olusegun Bankole, 22 May 2009

Contents

1. Introduction

1.1 Motivation for Study

The study of thermal decomposition of a viscous reactive fluid through parallel media is paramount in understanding the hydrodynamics of engineering processes [11]. Most generalised Newtonian fluids used in engineering and industrial processes become reactive during flow (i.e. when fluid is subjected to shear some of the work done is dissipated as heat) this makes them thermally decomposable e.g hydrocarbons and esters. Heat is generated internally within the fluid causing an increase in temperature within the fluid, the chemical reactions within the fluid follows mainly from the Arrhenius kinetics. Fluid properties such as viscosity is dependent on the temperature variation from time to time. Fluids in which the viscosity additionally depends on applied shear rates are called generalised Newtonian fluids . In this project we model this physical phenomena through a plane Couette flow, that is, flow between two parallel plates with the upper plate moving and the lower plate kept stationary. The fluid flow in the channel is thus driven only by the motion of the upper plate. A constant pressure is maintained within the channel. A lot of work has been done in the study of steady state flows problems e.g. [11, 12], where by steady flow we mean a fully developed flow in which the velocity, temperature etc. no longer depend on time. Currently, due mainly to well developed computational solution methods, attention has largely shifted to the study of unsteady flow problems.

Unsteady flows are gradually becoming the main focus of attention in computational rheology studies. The fast development of polymeric materials and associated polymer processing operations in the past decades has evoked significant research endeavors on the study of non-Newtonian fluid mechanics [10, 13]. The flow and motion of viscous reactive fluid between two plates occurs mostly in engineering and science applications. This process occurs practically in diverse areas e.g. in thermal applications — heat exchangers, cooling and drying technology, heating and air-conditioning techniques, in medical applications — subdomains of biomedicine and medical engineering, in chemical reactions — polymer melts, polmerisation reactor, catalyzers and in geological applications — extraction of oil and gas in geological fields [3].

The objectives of this project are to:

- Document the effects of variable viscosity in the thermal decomposition of relevant liquids subjected to unsteady one-dimensional shear flow and chemical reactions.
- Determine whether the particular constitutive modelling of viscosity (e.g via the Power Law, Carreau Model) play any significant role in such thermal decomposition.

The appropriate governing equations for the fluid flow will be solved via semi-implicit finite-difference schemes. A similar investigation was recently carried out on constant viscosity elastic fluids, see Chinyoka [2] it was demonstrated that fluid elasticity progressively delayed the onset of thermal decomposition. Possible future work thus arises via the extension to variable viscosity elastic liquids. We start off with a brief discussion of the characterisation of fluids as either Newtonian or non-Newtonian.

1.2 Newtonian Fluids

A Newtonian fluid is a fluid in which the stress is directly proportional to the velocity gradients. Newtonian fluids describe the physical properties that affect the stresses developed inside the fluid due to its motion and thus enters the dynamics of a flow. We explain what is meant by a Newtonian fluid in a simple shematic diagram in Figure 1.1. The fluid flows in the x-direction but with a velocity that varies in the y-direction. We consider a 1D flow hence the only non-zero component of the velocity is the x-component where u is a function of y $i.e.$ $u = u(y)$. Let PQ be an arbitrary plane perpendicular to y, across the plane PQ a stress acts. From Figure 1.1 the faster fluid above the plane PQ causes a forward drag in the fluid below and the slower fluid below will drag the fluid above back. Equal and opposite forces acts on the fluid above and below. The induced internal stress is termed viscous action. This stress is directly proportional to the velocity gradient. Mathematically we have the relation

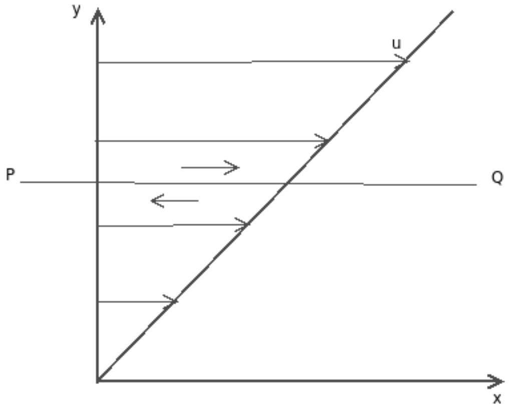


Figure 1.1: Viscous Stress Generated by Velocity Variation

$$\tau = \mu \frac{\partial u}{\partial y},$$

where τ is the force per unit area, μ the coefficient of viscosity of the fluid. The viscosity of a Newtonian fluid usually depends on the temperature, pressure and also the chemical composition of the fluid but does not depend on the forces acting on it. Examples of Newtonian fluids are water and air. Thus under constant temperature, constant chemical composition and constant pressure conditions, the viscosity of a Newtonian fluid is taken as constant. In the cartesian coordinate system, the equation governing the shear stress [7, 15] is

$$\tau_{ij} = -p\delta_{ij} + \mu\sigma_{ij},$$

with the deformation tensor σ_{ij} given by

$$\sigma_{ij} = \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right),$$

Physically, the tensor notation τ_{ij} is the shear stress in a fluid element on a face perpendicular to the i^{th} direction and acting in the j^{th} direction, u_i is the velocity in the i^{th} direction and x_j is the j^{th} direction co-ordinate, δ_{ij} the dirac delta symbol and p is the pressure field.

1.3 Generalised Newtonian Fluids

Any fluid that deviates from the above description is called a non-Newtonian fluid. These fluids have structural features which are influenced by the flow; this behaviour can arise in liquids with long molecules (polymers) and in liquids where the molecules tends to gather in more organised structures than usual. A distinguishing characteristic of various non-Newtonian fluid is the dependence of the apparent viscosity on the rate of strain. The dependence of the apparent viscosity on the rate of strain is a major concern in many applications and it can be modelled using fairly a straightforward modifications of the constitutive equation for a Newtonian fluid, the resulting equations describe the generalized Newtonian fluids. A generalized Newtonian fluid differs from a Newtonian fluid only in that the viscosity exhibits shear rate dependence while in Newtonian fluids viscosity is constant. We should however remark that the viscosity of any fluid additionally depends on the temperature, pressure and chemical composition.

1.3.1 Shear Thinning and Shear Thickening

Shear thinning is an effect where viscosity decreases with increasing rate of shear stress, fluids that possess this property are called pseudoplastic. This property of shear thinning occurs mostly in complex solutions such as ketchup, blood, paint and polymer melts—molten polysterene, polymer solutions—polyethylene oxide in water. It is observed that when paint is sheared with a brush, it flows easily but when the shear stress is removed the viscosity increases and no longer flows easily. On the other hand shear thickening is an effect where viscosity increases with the rate of shear, fluids with this property are termed dilatants. Dilatant effects occur when closely packed particles are combined with enough liquid to fill the gaps between them. At low velocities, the liquid acts as a lubricant, and the dilatant flows easily. At higher velocities, the liquid is unable to fill the gaps created between particles, and friction greatly increases, causing an increase in viscosity. This can readily be seen with a mixture of starch and water, which acts as solid when stabbed on the surface with a finger. Viscosity could be time dependent; when the viscosity increases with time at constant strain such fluids are called rheopectic fluids while the opposite behaviour when the viscosity decreases with time at constant strain are called thixotropic fluids. Some important types of non-Newtonian fluids are the viscoelastic fluids; these fluids behave both as elastic (solid) and viscous (fluids) and they exhibit memory effects in that the fluid can snap back to a previous configuration if stresses are removed. The viscoplastic fluid is a fluid which will not flow until a certain shear is applied. This shear stress must exceed a critical value known as the yield stress. So, viscoplastic fluids behave like solids when the applied shear stress is less than the yield stress, see Figure 1.2. Once the yield stress is exceeded, the viscoplastic fluid will flow like a fluid. In general, the viscosity of liquids decreases with temperature. Durst [3], explained that

- The viscosity of liquids decreases rapidly with temperature.
- The viscosity of gases increases with temperature at moderate pressure values.
- The viscosity of all liquids increases more or less with pressure.
- The pressure dependence of viscosity of gases is negligible.

In Chapter 2 we present the problem configuration and the governing equations satisfied by the fluid system with their associated boundary conditions, we proceed to non-dimensionalising these governing equations so that the solution of these equations could be generalised to any geometrically similar flow problems. In Chapter 3 we discuss the numerical method of solution employed and we also discuss

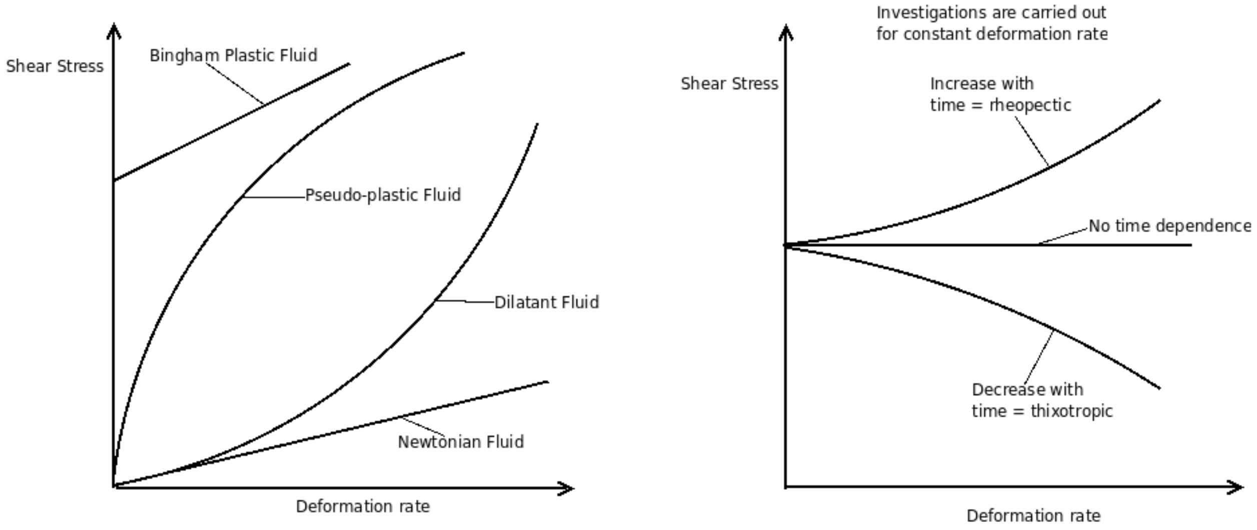


Figure 1.2: Properties of Newtonian and non-Newtonian Fluids [3].

the non-Newtonian models (Power Law and the Carreau model). We present and summarise the main results in Chapter 4.

2. Governing Equations

In this chapter we discuss the problem configuration, the governing equations together with the associated boundary conditions. We then proceed to non-dimensionalise the equations.

We consider the plane Couette flow in Figure 2.1 below

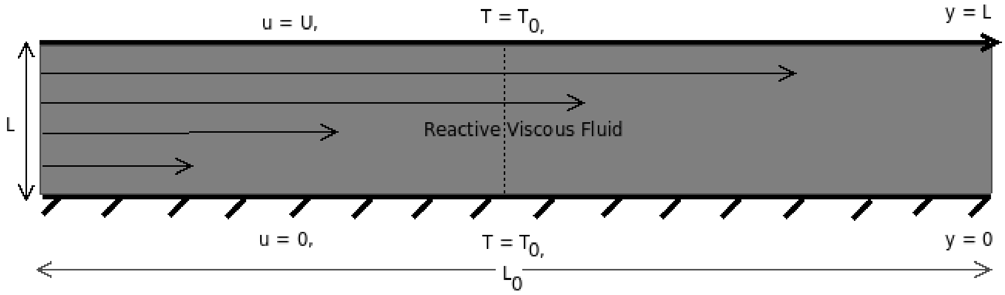


Figure 2.1: Problem Configuration

The geometry of the problem is such that the fluid is confined between two plates. The upper plate moves with velocity $u = U$ and the lower plate is stationary, $u = 0$. The boundary conditions are thus that the walls at $y = 0$, $y = L$ move with velocities 0 and U respectively and are maintained at the same temperature T_0. A viscous reactive fluid is enclosed between the plates and the velocity of the fluid is driven by the motion of the upper plate "a condition known as the no-slip boundary", the resulting velocity profile in a plane Couette flow, which predicts a linear velocity profile. As this fluid is sheared some of the workdone is dissipated as heat hence causing an increase in temperature within the fluid so we therefore assume that the heat is purely an internally generated heat, there exist no external heat through the plates (i.e. no suction or injection). We model this flow as 1D with velocity component $(u(t, y), 0, 0)$. In this model the ratio of the channel width to the channel length is $\ll 1$ giving a small aspect ratio that is $\epsilon = L/L_0 \ll 1$ (see Figure 2.1) where L_0 is the typical length scale of the channel and L is the typical width scale. The thermal conductivity, specific heat capacity and density of the liquid is represented by k, c_p and ρ respectively but will be treated as constant while the viscosity $\mu(T, \gamma)$ of the fluid is temperature and shear rate dependent. Table 2 shows some useful typical parameter notation [5].

2.1 Continuity Equation

The equation describing the conservation of scalar quantities such as mass M is obtained from the concept of a control volume. A control volume is any closed region in space. It is a region in which the rate of accumulation of mass is equal to the net rate at which the mass enters it by crossing the boundaries. The size and shape of a control volume may vary with time, and the control volume boundaries need not correspond to physical interfaces. Consider a fluid of density ρ, velocity $\mathbf{u}$ in an arbitrary control volume Ω, conservation of mass requires that

$$\int_\Omega \left(\frac{\partial}{\partial t} \rho + \nabla \cdot \mathbf{u} \rho \right) dV. = 0$$

Table 2.1: Typical Parameter and Notations

Symbol	Name	Unit
ρ	Fluid density	kgm^{-3}
μ_0	Typical dynamic viscosity	$kgm^{-1}s^{-1}$
μ	Dynamic viscosity	$kgm^{-1}s^{-1}$
ϵ	Aspect ratio of the flow	Dimensionless
U	Velocity scale	ms^{-1}
T	Temperature	K
ΔT	Temperature drop	K
t	Time	s
L	Channel height	m
L_0	Channel length	m
k	Thermal conductivity	$Wm^{-1}K^{-1}$
c_p	Heat capacity	$Jkg^{-1}K^{-1}$
$Re = \rho UL/\mu$	Reynolds number	Dimensionless
$Pe = \rho c_p LU/k$	Peclet number	Dimensionless
$Pr = \mu c_p/k$	Prandtl number	Dimensionless
$Br = \mu U^2/k\Delta T$	Brinkman number	Dimensionless

This holds for any arbitrary control volume Ω, Hence,

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

For incompressible fluid $\rho =$ constant and continuity equation reduces to

$$\nabla\cdot \mathbf{u} = 0. \tag{2.1}$$

2.2 Momentum Equation

The momentum equations are obtained from Newtons second law relating the total forces acting on a fluid element to the mass and acceleration of the fluid element. Neglecting body forces, the generalised momentum equations in cartesian coordinates are given by x, y and z momentum components [3].

x-momentum component is

$$\rho\frac{Du}{Dt} = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\left\{2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu(\nabla\cdot \mathbf{u})\right\} + \frac{\partial}{\partial y}\left\{\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right\} + \frac{\partial}{\partial z}\left\{\mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)\right\},$$

y-momentum component is

$$\rho\frac{Dv}{Dt} = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial y}\left\{2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu(\nabla\cdot \mathbf{u})\right\} + \frac{\partial}{\partial z}\left\{\mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right\} + \frac{\partial}{\partial x}\left\{\mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)\right\},$$

z-momentum component is

$$\rho\frac{Dw}{Dt} = -\frac{\partial P}{\partial z} + \frac{\partial}{\partial z}\left\{2\mu\frac{\partial w}{\partial z} - \frac{2}{3}\mu(\nabla\cdot \mathbf{u})\right\} + \frac{\partial}{\partial x}\left\{\mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)\right\} + \frac{\partial}{\partial y}\left\{\mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right\},$$

where μ is the coefficient of viscosity and P the pressure. In our case, flow is 1D that is **u** $= (u(t,y),0,0)$, and P is constant. The x-momentum equation reduces to

$$\rho\frac{Du}{Dt} = \frac{\partial}{\partial y}\left(\mu(T,\gamma)\frac{\partial u}{\partial y}\right),$$

where $\dfrac{D}{Dt} = \dfrac{\partial}{\partial t} + \mathbf{u}\cdot\nabla$ is the material derivative however $\mathbf{u}\cdot\nabla$ vanishes. Hence the momentum equation reduces to

$$\rho\frac{\partial u}{\partial t} = \frac{\partial}{\partial y}\left(\mu(T,\gamma)\frac{\partial u}{\partial y}\right). \tag{2.2}$$

with the corresponding boundary conditions $u(0) = 0$, $u(L) = U$.

2.3 Energy Equation

In the energy equation we include source terms based on the Arrhenius kinetics

$$\rho C_p\frac{DT}{Dt} = k\frac{\partial^2 T}{\partial y^2} + QC_0Ae^{-\frac{E}{RT}} + \mu(T,\gamma)\left(\frac{\partial u}{\partial y}\right)^2,$$

Making use of the definition of material derivative, we obtain the energy equation

$$\rho C_p\frac{\partial T}{\partial t} = k\frac{\partial^2 T}{\partial y^2} + QC_0Ae^{-\frac{E}{RT}} + \mu(T,\gamma)\left(\frac{\partial u}{\partial y}\right)^2, \tag{2.3}$$

with the boundary conditions $T(0) = T_0$ and $T(L) = T_0$.

where R the gas constant known as Boltzman gas constant, T the temperature in degree Kelvin, A the Arrhenius constant described as the pre-exponential factor (total number of collision leading to a reaction per second), E the activation energy which the colliding molecules must attain before a chemical reaction must take place, C_p is the heat capacity, k is the thermal conductivity, Q is the Heat flux and C_0 is the residual concentration of reactants. This reaction within the fluid is exothermic.

2.4 Dimensionless Equations

Equations (2.1), (2.2) and (2.3) are dimensional equations of motion governing the incompressible fluid in the narrow channel. They form a set of coupled, nonlinear, partial differential equations. The formidable problems encountered in solving this full set of non-linear equations is not made any easier by the number of parameters included in the problem. The number of parameters can be reduced by introducing dimensionless groups. We introduce the following dimensionless variables into equations (2.1), (2.2) and (2.3) to get the non-dimensional equations of motion [11]

$$x' = \frac{x}{L},\ \ y' = \frac{y}{L},\ \ u' = \frac{u}{U},\ \ t' = \frac{tU}{L},\ \ \mu(\Theta)' = \frac{\mu(\Theta)}{\mu_0}$$

The superscript $'$ represents dimensionless quantities.

$$\Theta' = \frac{E(T-T_0)}{RT_0^2},\ \ \lambda = \frac{QEAL^2C_0e^{-\frac{E}{RT_0}}}{RT_0^2k},\ \ \beta = \frac{\mu_0U^2e^{\frac{E}{RT_0}}}{QAL^2C_0},\ \ \varepsilon = \frac{RT_0}{E}$$

where λ, β, ε represent the Frank-Kamenetskii parameter, viscous heating parameter and the activation energy parameter respectively. The Frank-Kamenetskii parameter measures how reactive reactants are. The non-dimensional equations are obtained as follows:

- **Continuity equation**

$$\frac{\partial u'}{\partial x'} = 0$$

- **Momentun equation**

$$\begin{aligned} U\rho\frac{\partial(u'U)}{\partial(t'L)} &= \frac{\partial}{\partial(y'L)}\left(\mu(\Theta,\gamma)\frac{\partial(u'U)}{\partial(y'L)}\right) \\ \frac{\partial u'}{\partial t'} &= \frac{\mu_0}{\rho UL}\frac{\partial}{\partial y'}\left(\mu(\Theta,\gamma)'\frac{\partial u'}{\partial y'}\right) \end{aligned}$$

Hence the dimensionless equation for momentum is given as

$$\frac{\partial u'}{\partial t'} = \frac{1}{Re}\frac{\partial}{\partial y'}\left(\mu(\Theta,\gamma)'\frac{\partial u'}{\partial y'}\right) \tag{2.4}$$

- **Energy Equation**

Non-dimensionalising the LHS in equation (2.3) we obtain

$$\frac{\partial T}{\partial t} = \frac{RT_0^2U}{EL}\frac{\partial\Theta'}{\partial t'}$$

Non-dimensionalising the first term of RHS in equation (2.3) we obtain

$$\frac{\partial T^2}{\partial y^2} = \frac{RT_0^2U}{EL^2}\frac{\partial^2\Theta'}{\partial y'^2}$$

For the coupled velocity term in equation (2.3) we obtain

$$\left(\frac{\partial u}{\partial y}\right)^2 = \frac{U^2}{L^2}\left(\frac{\partial u'}{\partial y'}\right)^2.$$

For the reaction term in (2.3) we obtain

$$QC_0Ae^{-\frac{E}{RT}} = \frac{\lambda\varepsilon T_0k}{L^2}e^{\frac{\Theta'}{1+\varepsilon\Theta'}}.$$

Substituting into the energy equation (2.3) we have

$$\rho C_pU\frac{\partial\Theta'}{\partial t'} = \frac{k}{L}\frac{\partial^2\Theta'}{\partial y'^2} + \frac{\lambda k}{L}e^{\frac{\Theta'}{1+\varepsilon\Theta'}} + \mu(\Theta,\gamma)\frac{\lambda\beta k}{L}\left(\frac{\partial u'}{\partial y'}\right)^2.$$

Isolating $\frac{\partial\Theta'}{\partial t'}$ we obtain

$$\frac{\partial\Theta'}{\partial t'} = \frac{k}{\rho C_pUL}\frac{\partial^2\Theta'}{\partial y'^2} + \frac{\lambda k}{\rho C_pUL}e^{\frac{\Theta'}{1+\varepsilon\Theta'}} + \mu(\Theta,\gamma)\frac{\lambda\beta k}{\rho C_pUL}\left(\frac{\partial u'}{\partial y'}\right)^2,$$

$$Pe\frac{\partial \Theta'}{\partial t'} = \frac{\partial^2 \Theta'}{\partial y'^2} + \lambda e^{\frac{\Theta'}{1+\varepsilon\Theta'}} + \mu(\Theta,\gamma)Br\left(\frac{\partial u'}{\partial y'}\right)^2, \tag{2.5}$$

where $Pe = RePr = (\rho UL/\mu)(C_p\mu/k) = \rho C_p UL/k$ is the Peclet number, Re is the Reynolds number, Pr is the Prandtl number and Br is the Brinkman number

For convenience, we drop the $'$ superscript notation in (2.4) and (2.5).

$$\frac{\partial u}{\partial x} = 0, \tag{2.6}$$

$$Re\frac{\partial u}{\partial t} = \frac{\partial}{\partial y}\left(\mu(\Theta,\gamma)\frac{\partial u}{\partial y}\right), \tag{2.7}$$

$$Pe\frac{\partial \Theta}{\partial t} = \frac{\partial^2 \Theta}{\partial y^2} + \lambda e^{\frac{\Theta}{1+\varepsilon\Theta}} + \mu(\Theta,\gamma)Br\left(\frac{\partial u}{\partial y}\right)^2, \tag{2.8}$$

$\mu(\Theta,\gamma)$ is the dimensionless temperature and shear rate dependent viscosity.

Therefore equations (2.6), (2.7) and (2.8) corresponds to the dimensionless governing equations for the velocity u and temperature T with boundary conditions.

$$u(0) = 0, \;\; u(1) = 1, \tag{2.9}$$

$$\Theta(0) = \Theta(1) = 0. \tag{2.10}$$

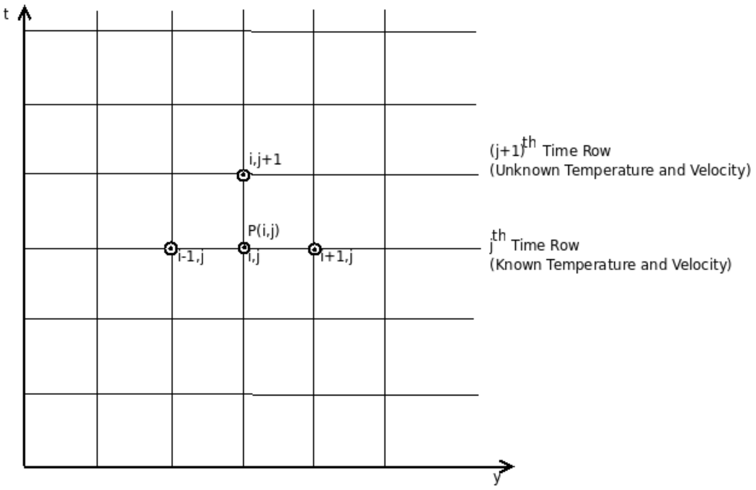
The following chapter introduces the numerical method of solution used in solving these coupled dimensionless equations, $(2.6 - 2.8)$ subject to boundary conditions $(2.9 - 2.10)$ and discusses the sensitivity of some parameters in the solution.

3. Numerical Solution and Discussion

In this chapter we discuss the finite-difference scheme used for the numerical integration of the coupled differential equations. We proceed to treat the the non-Newtonian models; the Power Law and the Carreau Model of viscosity and we explain the results obtained.

3.1 Finite-Difference Scheme

According to Gerald [6], finite-difference method is one of several methods for solving partial differential equations numerically. Continuous derivatives are replaced with discrete approximations which means the solution is known at a finite number of points. Generally, by increasing the number of these points, we improve and increase the accuracy of our solution.



t
i,j+1
(j+1)th Time Row
(Unknown Temperature and Velocity)
P(i,j)
i-1,j
i,j
i+1,j
jth Time Row
(Known Temperature and Velocity)
y

Figure 3.1: Grid on a Semi-infinite Strip

Definition 3.1.1. *A grid is a set of locations where the discrete solution is computed [6]*

The approximate solution to the partial differential equation is found at the points of intersection of these parallel lines, the points are called grid points or in other words (lattice, mesh, pivotal or nodal points). In this work we consider a uniform grid points through out, so the distance between the adjacent points in time and space are Δt and Δy respectively.

Our dimensionless equations contain derivatives with respect to time (t) and position (y). But as Δt and $\Delta y \to 0$ the numerical solution will approach the true exact solution. The rate at which numerical solution approaches the exact solution varies and depend on the scheme used. Exist two major schemes namely:

- Explicit finite-difference scheme;
- Implicit finite-difference scheme, further divided into fully implicit and semi-implicit schemes.

In this work we consider the semi-implicit scheme because it allows flexibility in values for $r = \Delta t/(\Delta y)^2$ (called the Courant number) and improves numerical stability.

3.2 Finite-Difference Approximation to Derivatives

We will consider the forward in time—center in space (FTCS) approximation to our derivatives.

3.2.1 Difference Formulas

First Order Forward Difference

By Taylor series expansion of $u(y)$ about the point y_i

$$\begin{aligned} u(y_i + \delta y) &= u(y_i) + \delta y \frac{\partial u}{\partial y}\bigg|_{y_i} + \frac{\delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} + \frac{\delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \dots \\ u(y_i - \delta y) &= u(y_i) - \delta y \frac{\partial u}{\partial y}\bigg|_{y_i} + \frac{\delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} - \frac{\delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \dots \end{aligned}$$

where $\delta y = \Delta y$, we obtain

$$\begin{aligned} u(y_i + \Delta y) &= u(y_i) + \Delta y \frac{\partial u}{\partial y}\bigg|_{y_i} + \frac{\Delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} + \frac{\Delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \dots \\ u(y_i + \Delta y) &= u(y_i) - \Delta y \frac{\partial u}{\partial y}\bigg|_{y_i} + \frac{\Delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} - \frac{\Delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \dots \end{aligned}$$

Solving for $\frac{\partial u}{\partial y}\bigg|_{y_i}$ we obtain

$$\frac{\partial u}{\partial y}\bigg|_{y_i} = \frac{u(y_i + \Delta y) - u(y_i)}{\Delta y} - \frac{\Delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} - \frac{\Delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \dots$$

If we substitute the approximate solution, that is $u_i \approx u(y_i)$ and $u_{i+1} \approx u(y_{i+1})$ we obtain

$$\frac{\partial u}{\partial y}\bigg|_{y_i} \approx \frac{u_{i+1} - u_i}{\Delta y} - \frac{\Delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} - \frac{\Delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \dots$$

So we have,

$$\frac{\partial u}{\partial y}\bigg|_{y_i} = \frac{u_{i+1} - u_i}{\Delta y} + O(y) \tag{3.1}$$

Equation (3.1) is the forward difference formula for $(\partial u/\partial y)_{y_i}$ because it involves nodes y_i and y_{i+1}

First Order Central Difference

The Taylor series expansions of u_{i+1} and u_{i-1}

$$\begin{aligned} u_{i+1} &= u_i + \Delta y \frac{\partial u}{\partial y} + \frac{\Delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} + \frac{\Delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \ldots, \qquad (3.2) \\ u_{i-1} &= u_i - \Delta y \frac{\partial u}{\partial y} + \frac{\Delta y^2}{2!}\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} - \frac{\Delta y^3}{3!}\frac{\partial^3 u}{\partial y^3}\bigg|_{y_i} + \ldots. \qquad (3.3) \end{aligned}$$

Substracting (3.3) from (3.2), we obtain

$$u_{i+1} - u_{i-1} = 2\Delta y \frac{\partial u}{\partial y}\bigg|_{y_i} + \frac{2(\delta x)^3}{3!}\frac{\partial^3 u}{\partial y^3} + \ldots$$

Solving for $(\partial u/\partial y)_{y_i}$, we obtain

$$\frac{\partial u}{\partial y}\bigg|_{y_i} = \frac{u_{i+1} - u_{i-1}}{2\Delta y} + O(y) \qquad (3.4)$$

Equation (3.4) is the central difference approximation to $(\partial u/\partial y)_{y_i}$

Second Order Central Difference

If we manipulate the Taylor series expansion about $u(y_i)$, by adding (3.3) and (3.2) we obtain

$$u_{i+1} + u_{i-1} = 2u_i + (\delta y)^2 \frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} + \frac{2(\delta x)^4}{4!}\frac{\partial^4 u}{\partial y^4}\bigg|_{y_i} + \ldots$$

Solving for $(\partial^2 u/\partial y^2)_{y_i}$ we obtain

$$\frac{\partial^2 u}{\partial y^2}\bigg|_{y_i} = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta y^2} + O(y). \qquad (3.5)$$

This is the central difference approximation to the second derivative.

Numerical Algorithm

Our numerical algorithm caters for all the three types of schemes namely explicit, semi-implicit and fully implicit, each scheme is implemented by varying the value of α, in the range $0 \le \alpha \le 1$. The discretisation of our governing equation is based on a rectangular Cartesian mesh and uniform grid on which the finite-differences are taken. The time derivatives are taken to be forward in time while the spatial derivatives are taken to be centre in space. Since the solution at the boundaries are known, the equations corresponding to the first and last grid point are adjusted to cater for the boundary conditions. Let us consider and discretise the general 1D heat parabolic equation making use of the forward in time and centre in space, we obtain

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta y)^2} + (1-\alpha)\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta y)^2} \qquad (3.6)$$

Equation (3.6) allows for the robust variation of α

- When $\alpha = 0$ the scheme reduces to the fully explicit scheme
- When $\alpha = \frac{1}{2}$ the scheme reduces to the Crank Nicolson scheme
- When $\alpha = 1$ the scheme reduces to the fully-implicit scheme

Where $(n + \frac{1}{2})$ is obtained by taking the average of $(n + 1)$ and (n)

Rearranging (3.6) and collecting the explicit terms to the right hand side to obtain the unknown terms we have

$$-\alpha r u_{i+1}^{n+1} + (1 + 2\alpha r)u_i^{n+1} - \alpha r u_{i-1}^{n+1} = (1-\alpha)r u_{i+1}^n + (1 - 2(1-\alpha)r)u_i^n + (1-\alpha)r u_{i-1}^n \tag{3.7}$$

where $r = \Delta t/(\Delta y)^2$. If we apply this scheme to (2.7) and (2.8) for both velocity and energy (temperature) equation we obtain

$$\begin{aligned} -\alpha r\mu^{(n)}u_{i+1}^{n+1} + (Re + 2\alpha r\mu^{(n)})u_i^{n+1} - \alpha r\mu^{(n)}u_{i-1}^{n+1} = (1-\alpha)r\mu^{(n)}u_{i+1}^n + \\ (Re - 2(1-\alpha)r\mu^{(n)})u_i^n + (1-\alpha)r\mu^{(n)}u_{i-1}^n + \\ \frac{r}{4}\left(u_{i+1}^n - u_{i-1}^n\right)\left(\mu_{i+1}^{(n)} - \mu_{i-1}^{(n)}\right) \end{aligned} \tag{3.8}$$

$$\begin{aligned} -\alpha r\Theta_{i+1}^{n+1} + (Pe + 2\alpha r)\Theta_i^{n+1} - \alpha r\Theta_{i-1}^{n+1} = (1-\alpha)r\Theta_{i+1}^n + \\ (Pe - 2(1-\alpha)r)\Theta_i^n + (1-\alpha)r\Theta_{i-1}^n + \Delta t\lambda \exp\left(\frac{\Theta_i^n}{1+\varepsilon\Theta_i^n}\right) + \Delta t\mu^{(n)}Br\left(\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta y}\right)^2 \end{aligned} \tag{3.9}$$

where the left-hand-side of (3.8) and (3.9) are tridiagonal matrices given as

$$A = \begin{pmatrix} Re + 2\alpha r\mu^{(n)} & -\alpha r & 0 & 0 & 0 & 0 \\ -\alpha r & Re + 2\alpha r\mu^{(n)} & -\alpha r & 0 & 0 & 0 \\ 0 & -\alpha r & Re + 2\alpha r\mu^{(n)} & -\alpha r & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -\alpha r & Re + 2\alpha r\mu^{(n)} & -\alpha r \\ 0 & 0 & 0 & 0 & -\alpha r & Re + 2\alpha r\mu^{(n)} \end{pmatrix}$$

$$B = \begin{pmatrix} Pe + 2\alpha r & -\alpha r & 0 & 0 & 0 & 0 \\ -\alpha r & Pe + 2\alpha r & -\alpha r & 0 & 0 & 0 \\ 0 & -\alpha r & Pe + 2\alpha r & -\alpha r & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -\alpha r & Pe + 2\alpha r & -\alpha r \\ 0 & 0 & 0 & 0 & -\alpha r & Pe + 2\alpha r \end{pmatrix}$$

for the velocity and temperature respectively.

The linear system i.e. $Au^{n+1} = b^n$ corresponding to the velocity is given as

$$\begin{pmatrix} Re + 2\alpha r & -\alpha r & 0 & 0 & 0 & 0 \\ -\alpha r & Re + 2\alpha r & -\alpha r & 0 & 0 & 0 \\ 0 & -\alpha r & Re + 2\alpha r & -\alpha r & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -\alpha r & Re + 2\alpha r & -\alpha r \\ 0 & 0 & 0 & 0 & -\alpha r & Re + 2\alpha r \end{pmatrix} \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ \vdots \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \end{pmatrix} = \begin{pmatrix} b_1^n \\ b_2^n \\ \vdots \\ \vdots \\ b_{N-2}^n \\ b_{N-1}^n \end{pmatrix}$$

The linear system $B\Theta^{n+1} = g^n$ corresponding to the temperature is given as

$$\begin{pmatrix} Pe+2\alpha r & -\alpha r & 0 & 0 & 0 & 0 \\ -\alpha r & Pe+2\alpha r & -\alpha r & 0 & 0 & 0 \\ 0 & -\alpha r & Pe+2\alpha r & -\alpha r & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & -\alpha r & Pe+2\alpha r & -\alpha r \\ 0 & 0 & 0 & 0 & -\alpha r & Pe+2\alpha r \end{pmatrix} \begin{pmatrix} \Theta_1^{n+1} \\ \Theta_2^{n+1} \\ \vdots \\ \vdots \\ \Theta_{N-2}^{n+1} \\ \Theta_{N-1}^{n+1} \end{pmatrix} = \begin{pmatrix} g_1^n \\ g_2^n \\ \vdots \\ \vdots \\ g_{N-2}^n \\ g_{N-1}^n \end{pmatrix}$$

where b^n and g^n are the explicit terms of the velocity and temperature respectively. Equations (3.8) and (3.9) are solved by inversion of the tridiagonal matrices to obtain the solutions $\mathbf{u}^{n+1}$ and Θ^{n+1}.

3.3 Power Law

This is an important rheological model which explains the relationship between the viscosity and shear rates, it also explains the relationship between the shear stress and shear rates over a range of shear rates where shear thinning occurs in non-Newtonian fluids. Apparent viscosity is defined as the ratio of shear stress to rate of shear. From Figure 1.2 it should be noted that for shear thinning and thickening, the shear stress-shear rate curve passes through the origin. This type of behaviour is often approximated by the 'power law' and such fluids are called 'power law fluids'. The power law, also called the Ostwald-de Waele is written as

$$\tau = K\gamma^n \tag{3.10}$$

where γ is the deformation rate $\frac{du}{dy}$. But for all power law non-Newtonian fluids, we can reexpress this equation to look more like a Newtonian fluid and from here we can observe some interesting classes of materials similar to dilatant and pseudoplastic properties. If we rewrite the stress

$$\tau = K|\gamma|^{n-1}\gamma = \eta_a\gamma \tag{3.11}$$

the power law viscosity becomes

$$\eta_a = K\gamma^{n-1} \tag{3.12}$$

where the apparent viscosity $\eta_a = \frac{\tau}{\gamma}$, the power n is called the power law index or the behaviour index and K is the consistency index, the units of K depends on the value of n . The power n gives the following physical interpretation:

- The shear thinning behaviour corresponds to $n < 1$
- The special case $n = 1$ corresponds to the Newtonian behaviour
- The shear thickening behaviour corresponds to $n > 1$

in this case the consistency index K corresponds to the viscosity μ in the Newtonian case. The values of n for shear thinning often extend to 0.5 but less commonly can be as 0.3 or even 0.2, while values for shear thickening behaviour usually extend to 1.2 or 1.3 [14]. But there exist one great limitation of the power law model, for zero shear rates the power law gives an infinite viscosity and at the same time it breaks down (fails) at high shear rates — where the viscosity ultimately approach a constant value—say zero value. In Figure 3.2(a) we see how the power law behaves at zero shear rates and infinite shear rates. Due to this shortcoming of the Power law we relax the model in this work. Nevertheless, the power law is still useful for modelling blood, polymers, rubber solutions etc.

3.4 Carreau Model

Another interesting model in non-Newtonian fluids is the Carreau model, it is similar to the power law model for intermediate values of shear rates. The Carreau model is bounded for both zero and infinite shear rates. Mathematically

$$\mu(\gamma) = \mu_\infty + (\mu_0 - \mu_\infty)\left[1 + (\beta\gamma)^2\right]^{\frac{n-1}{2}} \qquad \text{(isothermal case)} \tag{3.13}$$

and for the temperature and shear rate dependent viscosity using the Carreau model we have

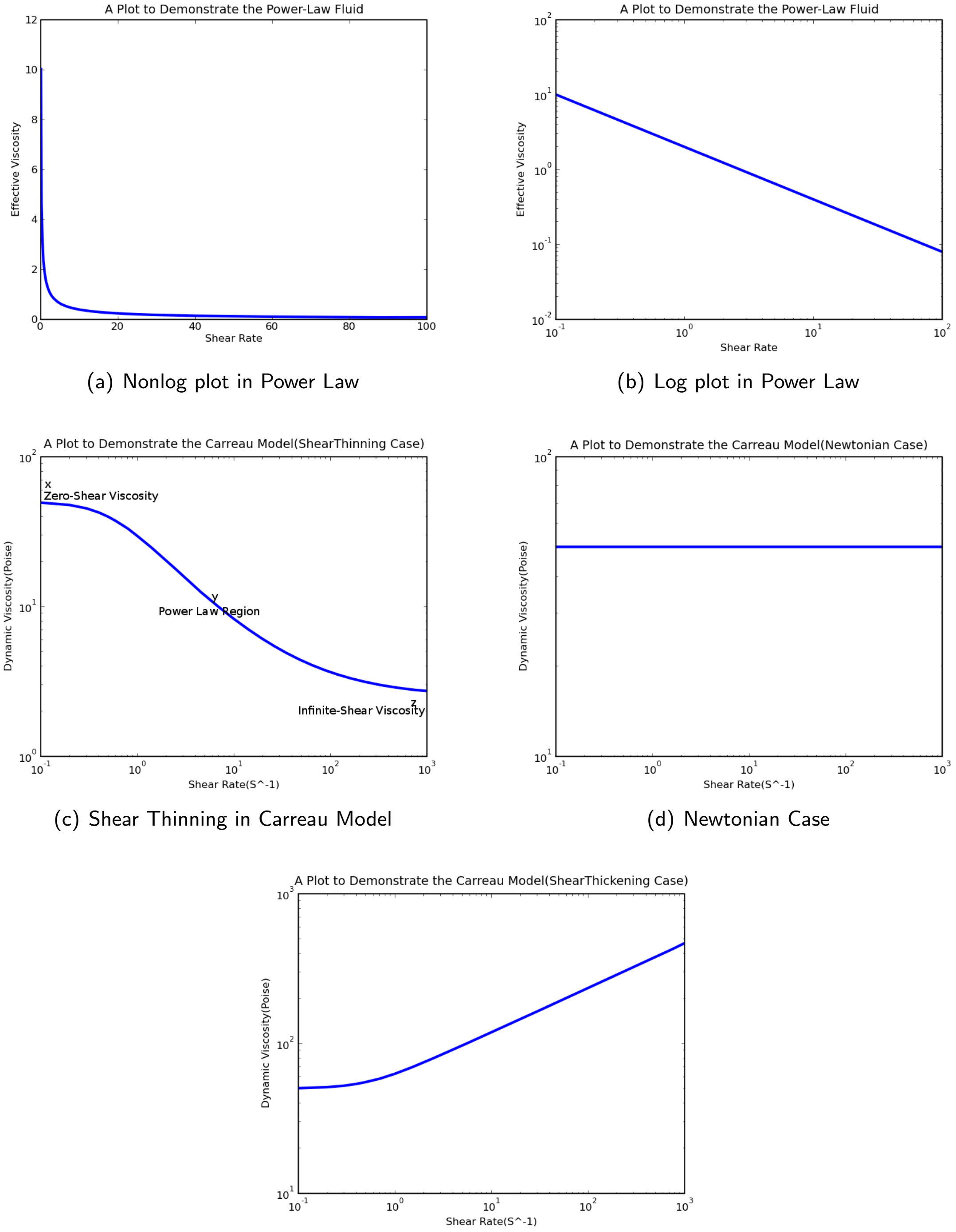
$$\mu(\Theta, \gamma) = \mu_\infty + (\mu_0 - \mu_\infty)\left[1 + (\beta\gamma)^2\right]^{\frac{n-1}{2}} e^{\frac{-\Theta}{1+\epsilon\Theta}} \qquad \text{(non-isothermal case)} \tag{3.14}$$

where the parameters n, β, μ_0, μ_∞ are always dependent on the fluid, λ is the time constant, μ_0, is the zero shear viscosity; μ_∞ is the infinite shear viscosity. Figure 3.2(c) below shows how the viscosity is bounded by μ_0 and μ_∞ at low and high shear rates. For example in polymeric fluids, viscosity dependence on shear rate can be shown by the plots in Figure 3.2 below having these properties:

- $\lim\limits_{\gamma\,\to\,0^+} \mu = \mu_0$
- As depicted in 3.2(c) in order to show how the viscosity behaves to its limiting value, while also using high shear rate behaviour, it is usually better that a log(μ) versus log(γ) be plotted.
- $\mu(\gamma)$ is a decreasing function of γ. This behaviour is described as "pseudoplastic" behaviour.
- As depicted in 3.2(c) for sufficiently low shear rate values $(\log(\gamma) \leq x)$, the viscosity is independent on the shear rate i.e. the polymeric fluid exhibits Newtonian behaviour. When $x \leq$ log(γ) $\leq y$, the dependence of log(μ) on log(γ) is non-linear. Finally when $y \leq$ log(γ) $\leq z$, μ has power law dependence on γ and as γ increases above z the viscosity levels out and the fluid becomes Newtonian again.
- In 3.2(c) $\mu(\gamma)$ shows a horizontal asymptote, which for most polymeric fluids it is impossible to determine experimentally due to polymer degradation at high shear rates.
- Figure 3.2(d) describes the Newtonian case $(n = 1)$
- Figure 3.2(e) illustrates the case of shear thickening.

3.5 Code Validation

Considering equations (2.6) and (2.7), the solutions to these coupled non-linear partial differential equations are the fluid velocity u and temperature Θ. These equations are very difficult to deal with analytically this is due to variation in viscosity, non-linear source terms in equation (2.7) and their transient nature. If we run our scheme at $t = 0$ we will recover the initial conditions to our problem graphically as depicted in 3.3(a) and 3.3(b). If we consider the assumption that the viscosity is constant, equation (2.6) with the boundary condition in equation (2.8) we obtain the classical linear velocity profile $u(y) = y$ see Figure 3.3(c) which supports our intuition for a plane Couette flow. Similarly, if we have a non-changing reaction source terms in equation (2.7) under the boundary condition in equation (2.8)



(a) Nonlog plot in Power Law

(b) Log plot in Power Law

(c) Shear Thinning in Carreau Model

(d) Newtonian Case

(e) Shear Thickening

Figure 3.2: Plots of Effective Viscosity with Shear Rates in Power Law and Carreau Model

we obtain a parabolic temperature profile see Figure 3.3(d) having its maximum value $(\Theta_{max} = 0.1390)$ along the centerline of the narrow channel.

If we now assume the viscosity to be temperature dependent only $\mu(\Theta)$. Using the parameter values

$Re = Pe = Br = 1$, $\lambda = 0.1$, $\varepsilon = 0.01$, $\Delta y = 0.02$ and $\Delta t = 0.001$ we still obtain the linear velocity profile and the parabolic temperature profile. We observe that at these parameter values the constant viscosity gave the maximum temperature $\Theta_{max} = 0.1390$ and for the variable viscosity we observe that the maximum temperature decreases slightly to $\Theta_{max} = 0.1308$. Infact, the maximum temperature, Θ_{max} converges to 0.1308 for any value of $t \geq 0.6$ for the variable viscosity (temperature only). The computational speed is of order of few seconds as we refine the meshes on a standard computer.

3.6 Sensitivity in Parameters

3.6.1 Effects of Reynolds Number in Convergence

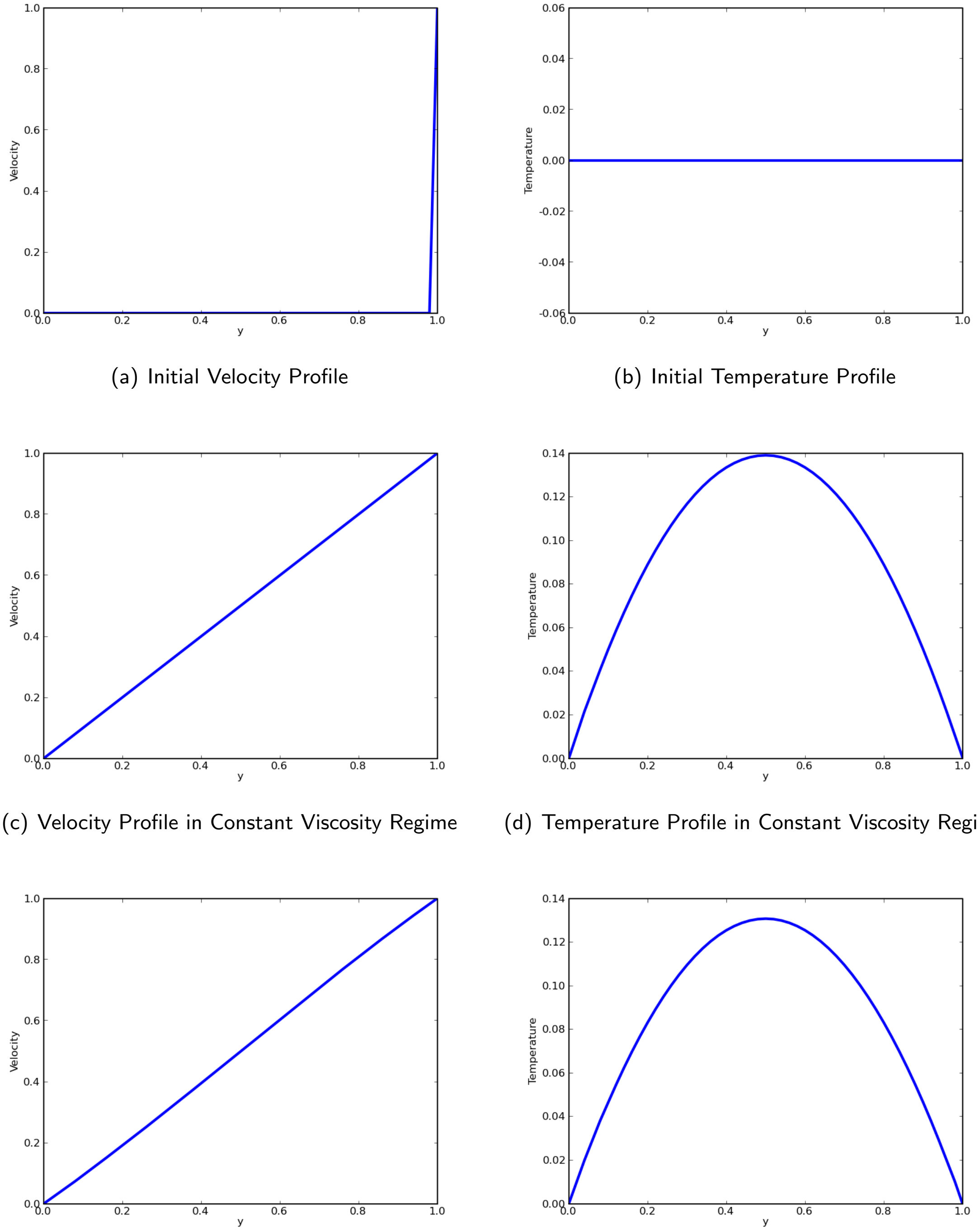
Reynolds number is a measure of of the relative importance of inertial and viscous effects in steady flows. Alternatively, it is a measure of the relative importance of the convective and diffusive transport of momentum [8]. Inertial (or convective) effects tends to be prominent when $Re \gg 1$ and nearly absent when $Re \ll 1$. In Figures 3.4(a) and 3.4(b) we obtain the developing velocity and temperature profiles at $t = 0.8$, but as we evolve in time all the profiles at different typical Reynolds number $Re = 1.0, 4.0, 7.0, 10.0$ converges to the same maximum teperature at $t = 20$ in Figures 3.4(c) and 3.4(d). The parameter values for these profiles are $Pe = Br = 1$, $\lambda = 0.1$, $\varepsilon = 0.01$, $\Delta y = 0.02$ and $\Delta t = 0.001$. Hence we say, the Reynolds number does not play a significant role in the thermal decomposition in our problem configuration and this is true if we consider the governing equations in the velocity equation (2.7), the Reynolds number is only a scalar multiplication of the time dependent velocity scale.

3.6.2 Effects of Peclet Number in Convergence

The Peclet number indicates the importance of convection relative to conduction or diffusion [8]. Alternatively, because Peclet number is the coefficient of the convective terms, it evidently shows the importance of convection relative to the molecular transport processes. If the Peclet number Pe is > 1 this implies the convective term is dominant and vice-versa. In Figure 3.5 we observed that the higher the Peclet number Pe the lesser the maximum temperature attained but we also observed that as we evolve in time to convergence the conduction terms balances the convective terms and the temperature field converges for all values of Pe to the same maximum temperature. The parameter values used are $Re = Br = 1$, $\lambda = 0.1$, $\varepsilon = 0.01$, $\Delta y = 0.02$ and $\Delta t = 0.001$ at $t = 0.8$ for the developing profile, we run it to convergence (steady state solution) at $t = 20$ for $Pe = 1.0, 3.0, 6.0, 9.0$. Similarly as the Reynolds number, in this range of values, the parameter does not play a significant role in the thermal decomposition.

3.6.3 Effects of Brinkman Number in Convergence

The Brinkman number Br expressses the relative importance of viscous dissipation and heat conduction. This occurs only in the energy equation, so higher values of the Brinkman parameter are expected to lead to an increase in the amount of heat generated within the fluid. Since this effect is caused by viscous dissipation (internal friction) the heat generated will be in form of mechanical heat as the fluid is sheared. This is shown in Figure 3.6. Figure 3.6(a) shows when the flow is not fully developed, we saw some variations in the velocity profile and the temperature field and Figure 3.6(b) shows when the fluid



(a) Initial Velocity Profile

(b) Initial Temperature Profile

(c) Velocity Profile in Constant Viscosity Regime

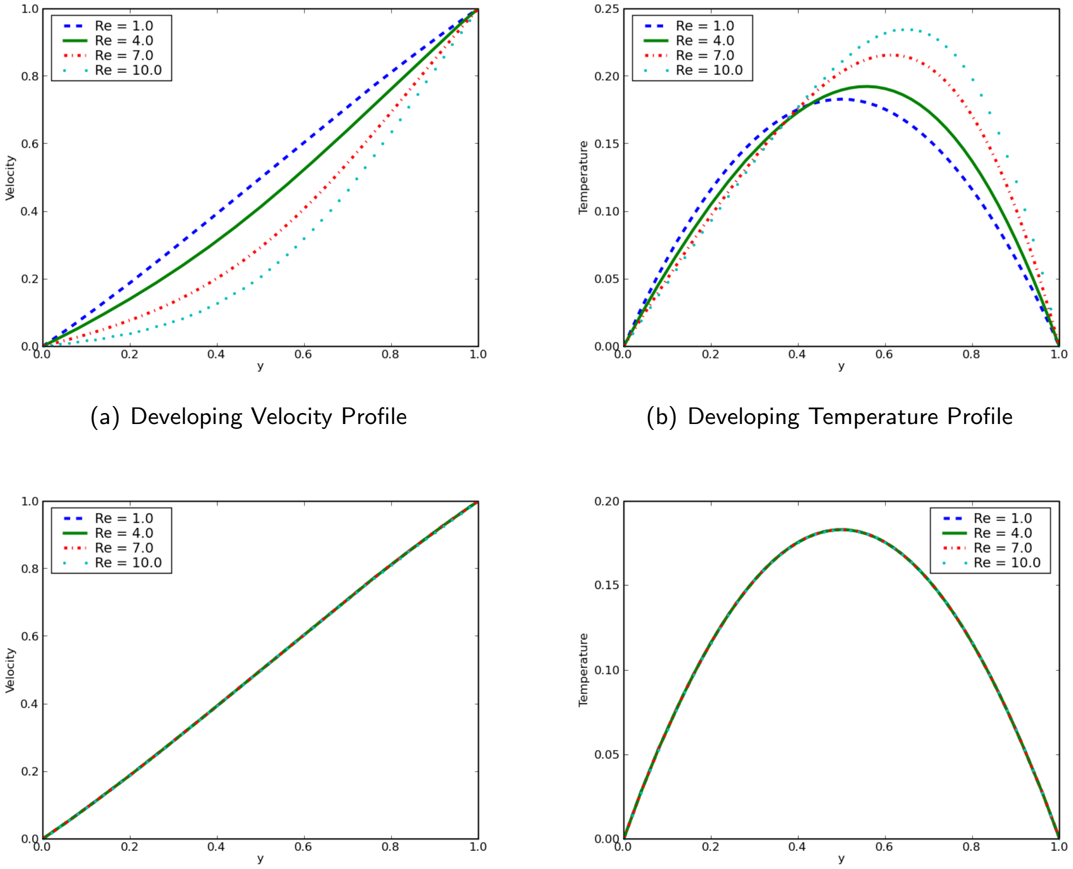
(d) Temperature Profile in Constant Viscosity Regime

(e) Velocity Profile in Variable Viscosity Regime

(f) Temperature Profile in Variable Viscosity Regime

Figure 3.3: Velocity and Temperature Profiles for Initial conditions and also as we Evolve in time

temperature field is now fully developed to convergence and we observed that the higher the Brinkmann number Br, the higher the maximum temperature which implies higher amount of heat is dissipated. This result is also in line with our intuition. The result is obtained when other parameters are fixed and we vary the Brinkman number only. The parameter values used are $Re = Pe = 1$, $\lambda = 0.1$, $\varepsilon = 0.01$,



(a) Developing Velocity Profile

(b) Developing Temperature Profile

(c) Fully Developed Velocity Profile

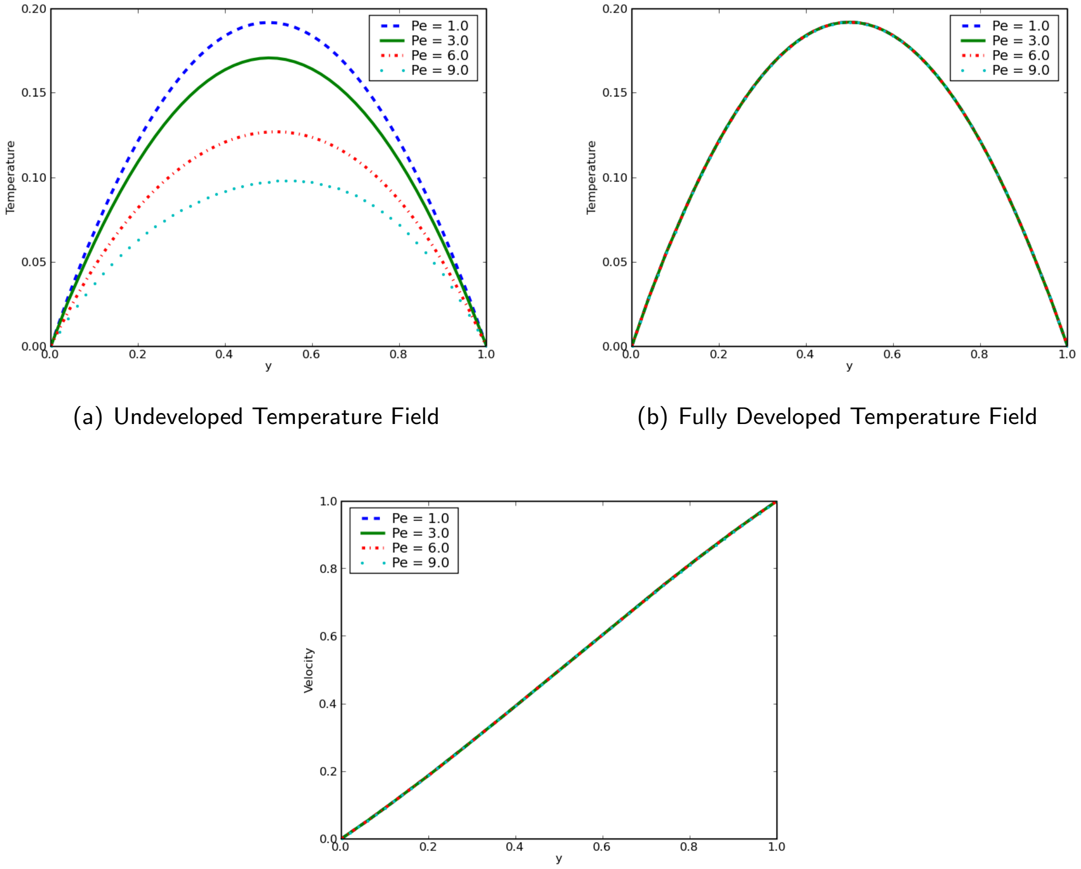
(d) Fully Developed Temperature Profile

Figure 3.4: Time Evolution in Reynolds Number

$\Delta y = 0.02$ and $\Delta t = 0.001$, $Br = 1.0, 4.0, 7.0, 10.0$ at $t = 0.8$

3.6.4 Reaction Parameter

If we look closely into the nature and structure of our governing equations in equations (2.6), (2.7) and (2.8) we should expect that the maximum temperature will increase as the reaction parameter λ increases, this is because an increase in this paramater will at the same time lead to an increase in the magnitude of source terms in the energy (temperature equation). In Figure 3.9 we keep track of the maximum temperature at different values of λ and we saw a long term behaviour of the fluid maximun temperature as we have higher values of λ. At these parameter values $Re = Pe = Br = 1$, $t = 0.6$ with λ as the independent variable $\varepsilon = 0.01$, $\Delta y = 0.02$ and $\Delta t = 0.001$, this result validates our intuition on the parameter λ.



(a) Undeveloped Temperature Field

(b) Fully Developed Temperature Field

(c) Fully Developed Velocity Profile

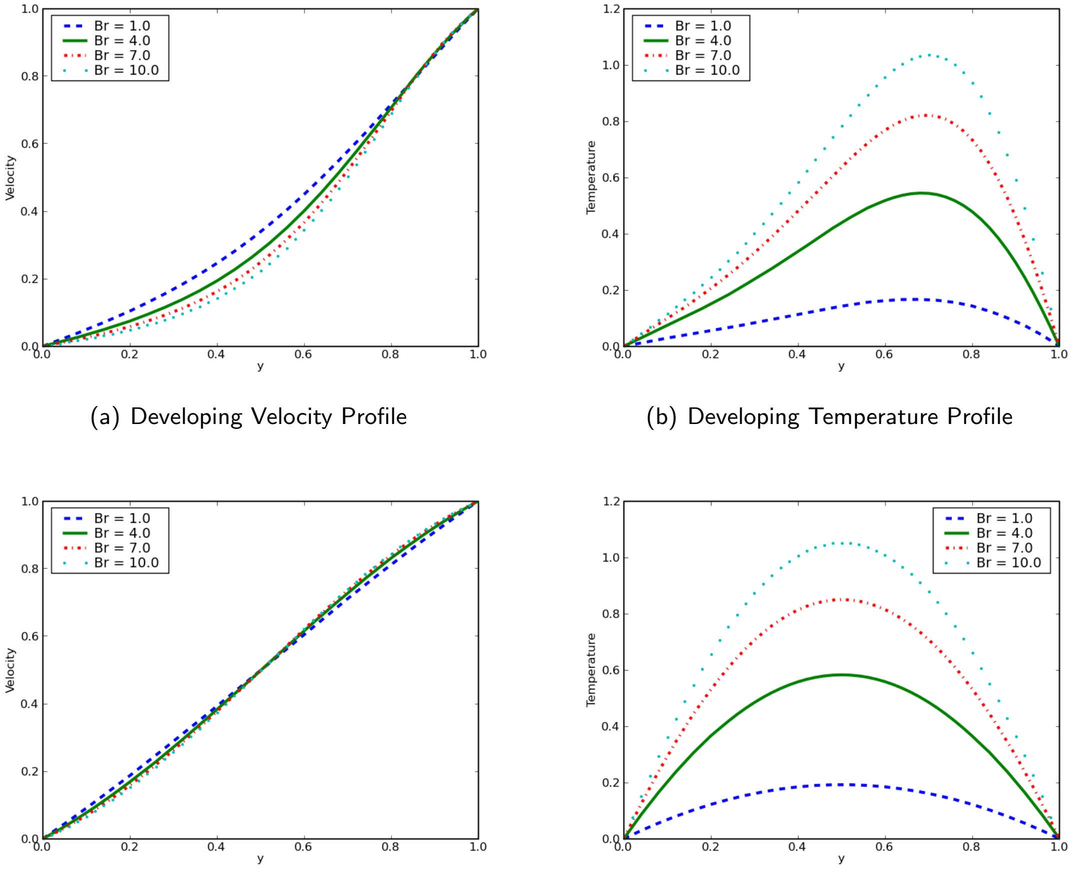
Figure 3.5: Time Evolution in Peclet Number

3.6.5 Variation in Time Constant

The parameter β is the characteristic time constant for the fluid. So $\beta\lambda$ in the Carreau model in (3.13) is the ratio of that time constant to a characteristic time for the flow. Figure 3.8(a) explains the dependence of temperature on the time constant β. It is observed that as the characteristic time constant β increases the maximum temperature decreases in magnitude and vice-versa.

3.6.6 Effects of Behaviour Index

In the Carreau Model in equation (3.13) we vary the behaviour index n in the shear thinning region $(i.e)$ when $n < 1$, Figure 3.10 illustrates the temperature field for both Carreau model and the power law, we observe that the power law temperature field converges to the same maximum temperature for values of $n < 1$. We observed that for $n = 1$ in the Carreau model we recover a highest maximun temperature compared to when $n < 1$. This implies that we obtain the highest maximum temperature for the Newtonian case and shear thinning case in generalised Newtonian fluids tends to reduce the rate of growth of the fluid maximum temperature. Plots of the maximum temperature with the reaction parameter values are shown in Figure 3.11. Figures 3.11(a) and 3.11(b) shows clearly that the threshold



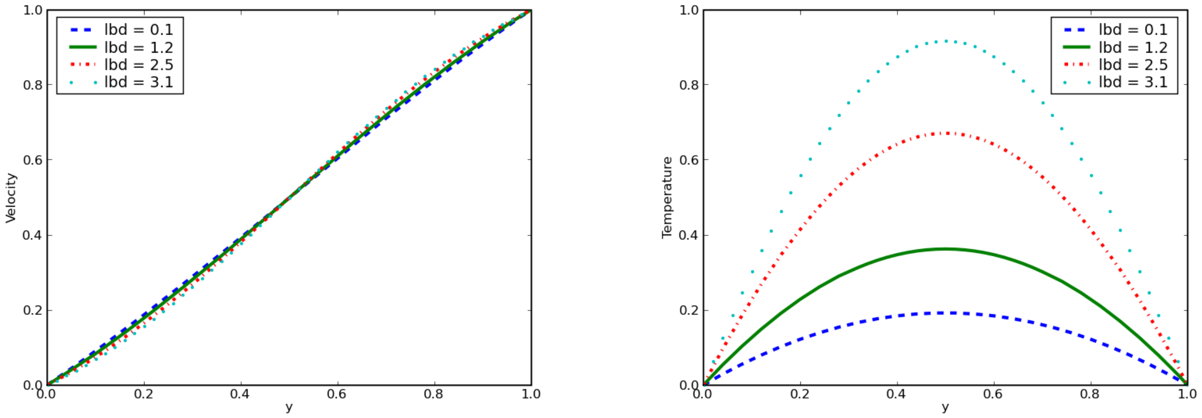
(a) Developing Velocity Profile

(b) Developing Temperature Profile

(c) Fully Developed Velocity Profile

(d) Fully Developed Temperature Profile

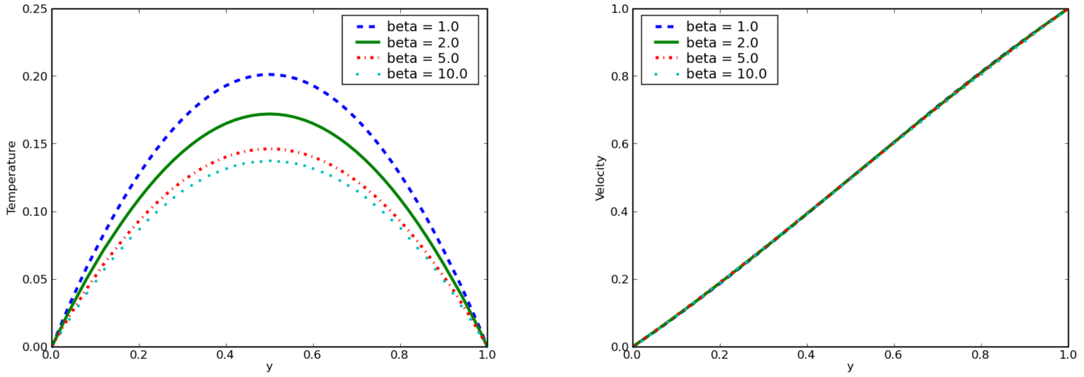
Figure 3.6: Time Evolution in Brinkman Number



(a)

(b)

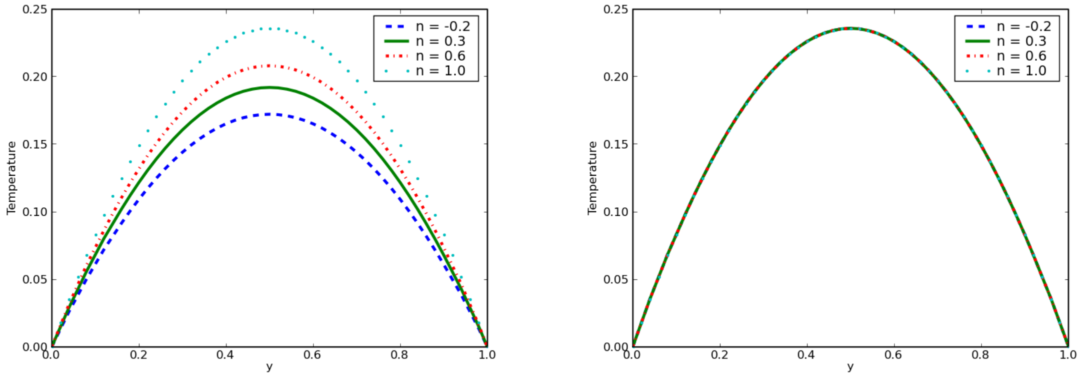
Figure 3.7: (a) Velocity Profile in Reaction Parameter Variation, (b) Temperature Profile in Reaction Parameter Variation



(a) Temperature Field

(b) Developed Velocity Field

Figure 3.8: Temperature and Velocity Profiles in the Time Constant



(a) Temperature Field in Carreau Law

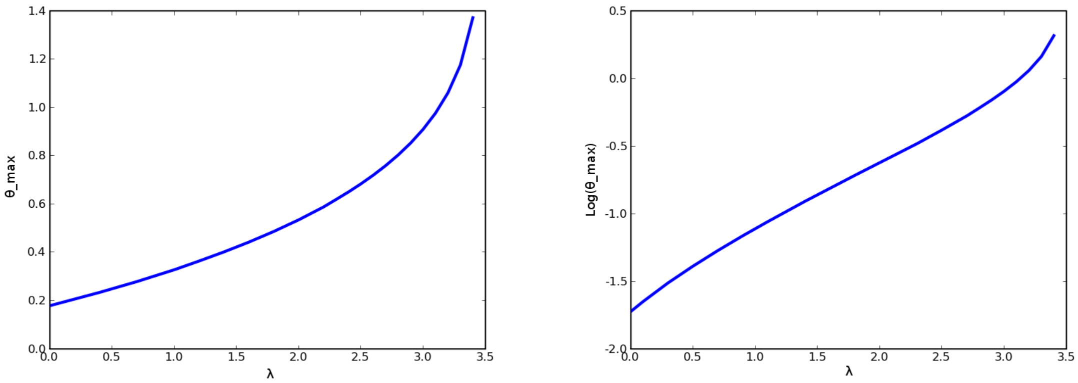
(b) Temperature Field in Power Law

Figure 3.9: Temperature Dependence on the behaviour index in both Carreau Model and Power Law

value of reaction parameter λ increases when we use generalised Newtonian fluids(GNF) in the shear thinning region. Generalised Newtonian fluids would therefore delay the onset of thermal runaway. Figures 3.11(a), 3.11(b), Figure 3.10 also show that generalised Newtonian fluids(GNF) have lower maximum temperatures when compared with Newtonian fluids. Thus shear thinning effects tend to lower the rate of increase of fluid maximum temperature.

3.6.7 Blow up of Solutions/Thermal Runaway

As we have shown in Figure 3.11, we obtain a long-term behaviour of the maximun temperature as parameter λ increases. But a blowup of solution may happen if the value of parameter λ exceeds a certain value (critical value) for example Chinyoka [2] stated that there could be bifurcation or blowup when the reaction parameter λ exceeds certain threshold values. In this work our numerical solution only caters for solutions for the fluid maximum temperature up to the point that corresponds to the critical values of λ. As we exceed the critical value our numerical algorithm cannot cater for the solution at

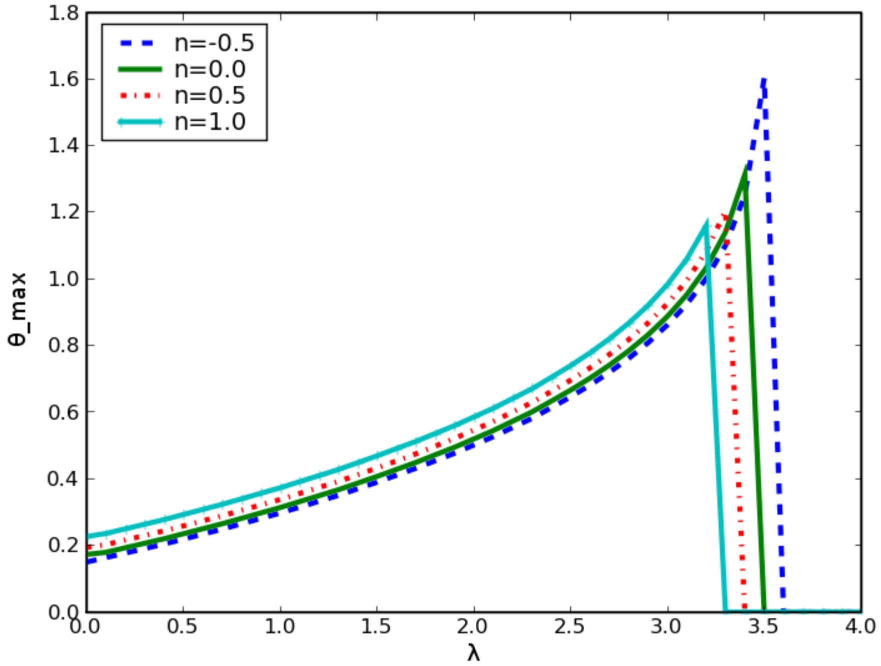


(a) Variation of Maximum Temperature with Lambda (b) Variation of Logarithm of Maximum Temperature with Lambda

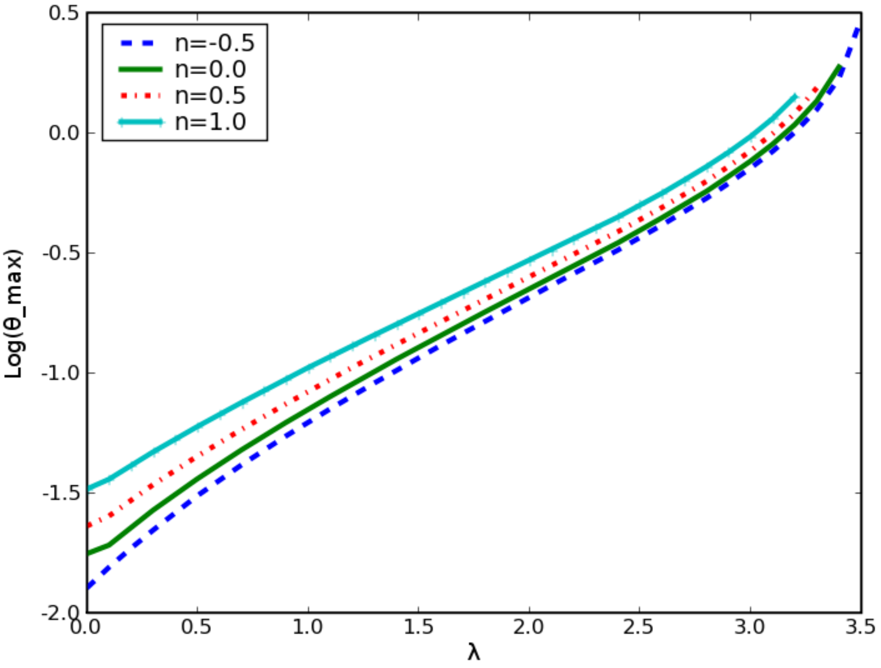
Figure 3.10: Variation of Maximum Temperature with respect to Reaction Parameter λ

that point hence we have a breakdown in our solutions as demonstrated in Figure 3.11. Figures 3.11(a) and 3.11(b) were obtained at the parameter values $Re = Pe = Br = 1$, $\varepsilon = 0.01$, $\Delta y = 0.02$ and $\Delta t = 0.001$, when $t = 0.5$. We vary values of n as $n = -0.5$ (dashed line), $n = 0.0$ (solid line), $n = 0.5$ (dash dot), $n = 1.0$ (light line).

The abrupt explosion of the physical temperature in finite time is termed Thermal Runaway. Thermal runaway is a situation where an increase in temperature causes a further increase in temperature, it could also be termed as a process by which an exothermic reaction goes out of control, often causing an explosion. Let us explain this process and its effects on equipments say gear lubricant. In the Aisoil Technical Service Bullettin [1] it was explained that extreme temperatures generated by vehicles increase stress on gear lubricants and can lead to a serious condition known as Thermal Runaway. As temperature increases the gear lubricants lose viscosity and load carrying capacity. When extreme loads breaks the lubricant film, there will be metal-to-metal contact, increasing friction and heat, in turn resulting in further viscosity loss, which further increases friction and heat. As heat continues to increase, viscosity decreases. Thermal Runaway is a cycle that leads to irreparable equipment damage from extreme wear. But in this work Thermal Runaway is exlusively ascribed to large increase in reaction parameter λ.



(a) Variation of Maximum Temperature with Lambda(Reaction Parameter) over some values of n



(b) Variation of Logarithm of Maximum Temperature with Lambda(Reaction Parameter) over some values of n

Figure 3.11: Thermal Criticalities

4. Conclusion

In this work we looked at the effects of variable viscosity in the thermal decomposition of fluids subjected to unsteady one-dimensional shear flow under Arrhenius kinetics. We employed the two constitutive modelling of viscosity: the power law and the Carreau model. We employed a zero initial condition and we fix the boundary conditions at the surfaces. The appropriate governing equation for the fluid are coupled, non-linear partial differential equations and because of their transient property we solved the equation via the semi-implicit finite-difference scheme. The power law model of viscosity is a deficient model because it produces an infinite viscosity at zero shear rate and a zero viscosity at high shear rate. Due to this shortcoming we dropped the power law in this work, we thus employed the Carreau model which allows for all values of shear rates: this model is bounded for both zero and infinite shear rates.

We performed a code validation for our scheme and we established some classical results of the plane Couette flow, we observed that the maximum attainable temperature for variable viscosity decreases slightly to 0.1308 when compared with the attainable temperature for the constant viscosity value of 0.1390 after running the code to convergence at say $t \geq 0.6$. We recovered the expected linear velocity profile and parabolic temperature profile with the maximum temperature along the center line. We considered the effects of some dimensionless parameter values in the thermal decomposition process. For the Reynolds number, we considered low Reynolds number in the range of values 1.0, 4.0, 7.0, 10.0. As we evolve in time to convergence each distinct temperature profiles for different Re values converges to the same profile, hence low Reynolds numbers do not play a crucial role in the decomposition process. We obtained similar results for the Peclet number Pe, but before convergence, a lower Peclet number gives a higher attainable temperature and vice-versa. We saw a significant thermal effect through varying the Brinkman number, we observed that even at steady state (fully developed) solution, higher values of Brinkman number gave a higher maximum attainable temperature, this results supports our intuition because the Brinkman number constitutes the source terms part in the heat generated internally within the fluid. For the reaction parameter λ, we demonstrated that this parameter plays a key role in the decomposition process, we saw that increase in it gave an increasingly long term behaviour in temperature. When λ exceeds a critical value λ_c our numerical algorithm fails to capture the solution. For values of λ greater than this critical value, we have thermal runaway. We obtained the critical value of 3.6 which is higher than the steady state value for a Newtonian problem, given by 2.6767, see Makinde [11]. We established that when the behaviour index $n < 1$ which implies the shear thinning region we have an increase in the critical values which eventually delay the onset of thermal runaway while in the Newtonian case $(n = 1)$ thermal runaway sets in earlier. In summary, in the shear thinning region of generalised Newtonian fluids we obtained,

- Decrease in attainable temperature in non-Newtonian nature of the fluid,
- Delay in thermal runaway which implies an increase in the critical value of the reaction parameter λ.

Conclusively, we demonstrated the superiority of non-Newtonian fluids typified by the Carreau model over Newtonian fluids in thermal decomposition, this result will be most useful in lubrication processes that involves thermally reactive fluids.

Appendix A. Numerical Codes Written in PYTHON

A.1 Code for Power Law Plot

```
"""
This function plots the Power Law Fluid:Dynamic viscosity against the Shear Rate
"""
import pylab as P
from scipy import log
from scipy import log10
from scipy import zeros
#Constants
K = 2.0  #Consistency Index
n  = 0.3 #Shear Thinning Behaviour
#n = 1.0 #Newtonian Behaviour
#n = 1.5 #Shear Thickening Behaviour
# main ============================================================
if __name__ == "__main__":
        #Array of Values of Shear Rate
        sr = P.arange(.1,100.0,.1)
        sr1 = sr**(n-1)
        #Viscosity to be Plotted
        mu = K*sr1
        #Plot Function
        P.title("A Plot to Demonstrate the Power-Law Fluid")
        P.loglog((sr),(mu),linewidth = 3)
        #P.plot((sr),(mu),linewidth = 3)
        P.xlabel("Shear Rate")
        P.ylabel("Effective Viscosity")
        P.show()
```

A.2 Code for Carreau Plot

```
"""
This function plots the Carreau Model:Dynamic viscosity against the Shear Rate
"""
import pylab as P
from scipy import log
from scipy import log10
from scipy import zeros
#Constants
```

```
mu_inf = 2.5
mu_0 = 50.0
lbd = 2.0 #Time Constant
n  = 0.3 #Shear Thinning Behaviour
#n = 1.0 #Newtonian Behaviour
#n = 1.5 #Shear Thickening Behaviour
# main =============================================================
if __name__ == "__main__":
        #Array of Values of Shear Rate
        sr = P.arange(.1,1000.0,.1)
        #Viscosity to be Plotted
        mu = mu_inf + (mu_0 - mu_inf)*(1 + (lbd*sr)**2)**((n-1)/2)
        #Plot Function
        P.title("A Plot to Demonstrate the Carreau Model(Non-Log Plot Case)")
        #P.loglog((sr),(mu),linewidth = 3)
        P.plot((sr),(mu),linewidth = 3)
        P.xlabel("Shear Rate(S^-1)")
        P.ylabel("Dynamic Viscosity(Poise)")
        P.show()
```

A.3 Code for Constant and Variable Viscosity

```
"""
This program simulates the Coupled Velocity and Energy
equations for Constant and Variable Viscosity
"""
###Imports###
from Numeric import *
import Gnuplot
from math import *
import matplotlib
from pylab import *
from scipy import *
from scipy.sparse import csc_matrix
from scipy.linsolve import spdiags, spsolve, use_solver
###Constants###
Re = 1.0
Pe = 1.0
Br = 1.0
lbd = 0.1
epsilon = 0.01
delta_t = 0.001
delta_y = 0.02
N = int(1/delta_y)
r = delta_t/(delta_y**2)
```

```
y = arange(0, 1.0+delta_y, delta_y)
##Initial Condition##
u = zeros(N+1)
T = zeros(N+1)
##Boundary Condition##
u[0] = 0
u[-1] = 1
T[0] = 0
T[-1] = 0
uold = u
Told = T
##Constants for the Carreau Model##
m = 2.0
n = 0.3 ##Behaviour Index##
beta = 2.0 ##Time Constant##
sr = 1   ##Shear rate##
alpha = 0.5  ##Controls the kind of scheme used##
K = 2.0 ##Consistency Index for Power law##
##Constants##
#muold = ones(N+1)
##Temperature only##
#muold = exp(-(Told/(1 + (epsilon*Told))))
#muold = (K/sr**(1-n))*exp(-(Told/(1 + (epsilon*Told))))#Power Law
##Carreau Model only##
#muold = (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))*ones(N+1)
##Viscosity dependence on Temperature and Shear rate##
muold = exp(-(Told/(1 + (epsilon*Told)))) * (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))
urhs = zeros(N+1)
Trhs = zeros(N+1)
##Matrix for Temperature##
vt = concatenate( ([Pe + 2*alpha*r,-alpha*r], zeros(N - 3) ), axis =0)
Mt = sparse.csc_matrix(linalg.toeplitz(vt,vt))
for i in range(20000): ##Controls the number of time steps##
        ##This loop solves the inner values##
        for i in range(1,N):
                urhs[i] = (Re- 2*(1-alpha)*r*muold[i])*uold[i] \
                + (1-alpha)*r*muold[i] * (uold[i+1] + uold[i-1]) \
                + (r/4) * (uold[i+1]-uold[i-1])*(muold[i+1]-muold[i-1])
                Trhs[i] = (Pe- 2*(1-alpha)*r) * Told[i] \
                + (1-alpha)*r *(Told[i+1] + Told[i-1]) \
                + delta_t*lbd*exp(Told[i]/(1 + epsilon*Told[i])) \
                + muold[i] * delta_t*Br*((uold[i+1]-uold[i-1])**2)/(4*delta_y**2)
        ##This expression caters for boundary conditions for velocity
        urhs[1] = urhs[1] + alpha * r * muold[1] * u[0]
        urhs[N-1] = urhs[N-1] + alpha * r * muold[N-1] * u[N]
        ##This expression caters for boundary conditions for temperature
        Trhs[1] = Trhs[1] + alpha * r * T[0]
        Trhs[N-1] = Trhs[N-1] + alpha * r * T[N]
        uurhs = urhs[1:N]
        ##Matrix for Velocity##
        vu = concatenate(([Re+2*alpha*r*muold[i],-alpha*r*muold[i]],zeros(N-3)),axis =0)
        Mu = sparse.csc_matrix(linalg.toeplitz(vu,vu))
        unew = linsolve.spsolve(Mu, uurhs)##Solves the Linear System##
        u[1:N] = unew
        uold = u##Prepares for the next iteration##
        TTrhs = Trhs[1:N]
        Tnew = linsolve.spsolve(Mt, TTrhs)##Solves the Linear System##
        T[1:N] = Tnew
        Told = T  ##Prepares for the next iteration##
        ##Constants##
        #muold = ones(N+1)
        ##Temperature only##
        #muold = exp(-(Told/(1 + (epsilon*Told))))
        ##Carreau Model only##
        #muold = (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))*ones(N+1)
        #muold = (K/sr**(1-n))*exp(-(Told/(1 + (epsilon*Told))))#Power Law
        ##Viscosity dependence on Temperature and Shear rate##
        muold = exp(-(Told/(1+(epsilon*Told))))*(1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))

print "The maximum Temperature is:=", max(T)
print "The values of u are =", u
print "The values of tmp are =", T
##Plotting Commands##
figure(1)
plot(y,T,linewidth = 3)
xlabel("y")
ylabel("Temperature")
figure(2)
plot(y,u,linewidth = 3)
xlabel("y")
ylabel("Velocity")
show()
```

A.4 Code for Variation of Parameters

```
"""
This program simulates the Coupled Velocity and Energy equations
in a plane Couette flow by varying some parameter values.
"""
###Imports###
from Numeric import *
import Gnuplot
from math import *
import matplotlib
```

```
from pylab import *
from scipy import *
from scipy.sparse import csc_matrix
from scipy.linsolve import spdiags, spsolve, use_solver

delta_t = 0.001
delta_y = 0.02
y = arange(0, 1.0+delta_y, delta_y)
def create_T(Pe, Re, Br, lbd, epsilon, n):

        """constants"""
        delta_t = 0.001
        delta_y = 0.02
        N = int(1/delta_y)
        r = delta_t/(delta_y**2)
        y = arange(0, 1.0+delta_y, delta_y)
        ##Initial Condition##
        u = zeros(N+1)
        T = zeros(N+1)
        ##Boundary Condition##
        u[0] = 0
        u[-1] = 1
        T[0] = 0
        T[-1] = 0
        uold = u
        Told = T
        ##Constants for the Carreau Model##
        m = 2.0
        beta = 2.0
        sr = 1
        K= 2.0
        #muold = (K/sr**(1-n))*exp(-(Told/(1 + (epsilon*Told))))##Power
        #muold = exp(-(Told/(1 + (epsilon*Told))))
        muold = exp(-(Told/(1 + (epsilon*Told)))) \
        *(1 + (m-1)*((1+(beta*sr)**2)**((n-1)/2))) ##Carreau Model##
        alpha = 0.5 ##Controls the scheme used##

        urhs = zeros(N+1)
        Trhs = zeros(N+1)
        ##Sparse Matrix for Temperature##
        vt = concatenate( ([Pe + 2*alpha*r,-alpha*r], zeros(N - 3) ), axis =0)
        Mt = sparse.csc_matrix(linalg.toeplitz(vt,vt))

        """ end of constants"""

        for i in range(20000):#Number of time steps
                for i in range(1,N):
                        urhs[i] = (Re- 2*(1-alpha)*r*muold[i])*uold[i] \
                        + (1-alpha)*r*muold[i] * (uold[i+1] + uold[i-1]) \
                        + (r/4) * (uold[i+1]-uold[i-1])*(muold[i+1]-muold[i-1])
                        Trhs[i] = (Pe- 2*(1-alpha)*r) * Told[i] \
                        + (1-alpha)*r *(Told[i+1] + Told[i-1]) \
                        + delta_t*lbd*exp(Told[i]/(1 + epsilon*Told[i])) \
                        + muold[i] * delta_t*Br*((uold[i+1]-uold[i-1])**2)/(4*delta_y**2)
                ##This expression caters for boundary conditions for velocity
                urhs[1] = urhs[1] + alpha * r * muold[1] * u[0]
                urhs[N-1] = urhs[N-1] + alpha * r * muold[N-1] * u[N]
                ##This expression caters for boundary conditions for temperature
                Trhs[1] = Trhs[1] + alpha * r * T[0]
                Trhs[N-1] = Trhs[N-1] + alpha * r * T[N]
                uurhs = urhs[1:N]
                ##Sparse Matrix for Velocity##
                vu=concatenate(([Re+2*alpha*r*muold[i],-alpha*r*muold[i]],zeros(N-3)))
                Mu = sparse.csc_matrix(linalg.toeplitz(vu,vu))
                unew = linsolve.spsolve(Mu, uurhs)
                u[1:N] = unew
                uold = u
                TTrhs = Trhs[1:N]
                Tnew = linsolve.spsolve(Mt, TTrhs)
                T[1:N] = Tnew
                Told = T
                ##Constants##
                #muold = ones(N+1)
                ##Temperature only##
                #muold = exp(-(Told/(1 + (epsilon*Told))))
                ##Carreau Model only##
                #muold = (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))*ones(N+1)
                #muold = (K/sr**(1-n))*exp(-(Told/(1 + (epsilon*Told))))#Power
                ##Viscosity dependence on Temperature and Shear rate##
                muold = exp(-(Told/(1 + (epsilon*Told)))) \
                * (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))##Carreau Model##

        print "The maximum Temperature is:=", max(T)
        print "The values of u are =", u
        print "The values of tmp are =", T
        return (T, u)
Re = 1.0
Pe = 1.0
Br = 1.0
lbd = 0.1
epsilon = 0.01
n = -0.5
(T, u) = create_T(Pe, Re, Br, lbd, epsilon, n)
n = 0.0
(T1, u1) = create_T(Pe, Re, Br, lbd, epsilon, n)
n = 0.5
```

```
(T2, u2) = create_T(Pe, Re, Br, lbd, epsilon, n)
n = 1.0
(T3, u3) = create_T(Pe, Re, Br, lbd, epsilon, n)
figure(1)
plot(y,T,'--', label='n =-0.5 ',linewidth = 3)
plot(y, T1,'-', label='n = 0.0 ',linewidth = 3)
plot(y, T2,'-.', label='n = 0.5 ',linewidth = 3)
plot(y, T3,'.', label='n = 1.0',linewidth = 3)
legend(loc='best')
xlabel("y")
ylabel("Temperature")
figure(2)
plot(y,u,'--', label='n = -0.5',linewidth = 3)
plot(y,u1,'-', label='n = 0.0',linewidth = 3)
plot(y,u2,'-.', label='n = 0.5',linewidth = 3)
plot(y,u3,'.', label='n = 1.0',linewidth = 3)
legend(loc='best')
xlabel("y")
ylabel("Velocity")
show()
```

A.5 Code for Thermal Runaway

```
"""
This program simulates the coupled velocity and energy equations and \
outputs the thermal runaway phenomenom
"""
###Imports###
from Numeric import *
import Gnuplot
from math import *
import matplotlib
from pylab import *
from scipy import *
from scipy.sparse import csc_matrix
from scipy.linsolve import spdiags, spsolve, use_solver
###Constants###
Re = 1.0
Pe = 1.0
Br = 1.0
#lbd = 3.0
epsilon = 0.01
delta_t = 0.001
delta_y = 0.02
N = int(1/delta_y)
r = delta_t/(delta_y**2)
```

```
y = arange(0, 1.0+delta_y, delta_y)
##Initial Condition##
u = zeros(N+1)
T = zeros(N+1)
##Boundary Condition##
u[0] = 0
u[-1] = 1
T[0] = 0
T[-1] = 0
uold = u
Told = T
##Constants for the Carreau Model##
m = 2.0
#n = 0.3
beta = 2.0
sr = 1 ##Shear rate
alpha = 0.5
##Constants##
#muold = ones(N+1)
##Temperature only##
#muold = exp(-(Told/(1 + (epsilon*Told))))
##Carreau Model only##
#muold = (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))*ones(N+1)
##Viscosity dependence on Temperature and Shear rate##
#muold = exp(-(Told/(1 + (epsilon*Told)))) * (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))
#lbdlist = arange(0,3.5,0.1)
#maxiTlist = zeros(len(lbdlist))
def Tmax(lbd,n):

        uold = u
        Told = T
        #muold = exp(-(Told/(1 + (epsilon*Told))))
        muold = exp(-(Told/(1 + (epsilon*Told)))) \
        * (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))
        urhs = zeros(N+1)
        Trhs = zeros(N+1)
        ##Sparse Matrix for Temperature##
        vt = concatenate(([Pe + 2*alpha*r,-alpha*r], zeros(N - 3)))
        Mt = sparse.csc_matrix(linalg.toeplitz(vt,vt))
        ##Iteration##
        #count = 0
        #while count < N:
        #count = count + 1
        for i in range(500):
                for i in range(1,N):
                        urhs[i] = (Re- 2*(1-alpha)*r*muold[i])*uold[i] \
                         + (1-alpha)*r*muold[i] * (uold[i+1] + uold[i-1]) \
                         + (r/4) * (uold[i+1]-uold[i-1])*(muold[i+1]-muold[i-1])
                        Trhs[i] = (Pe- 2*(1-alpha)*r) * Told[i] \
                         + (1-alpha)*r *(Told[i+1] + Told[i-1]) \
                         + delta_t*lbd*exp(Told[i]/(1 + epsilon*Told[i]))
                         + muold[i] * delta_t*Br*((uold[i+1]-uold[i-1])**2)/(4*delta_y**2)
                ##This expression caters for boundary conditions for velocity
                urhs[1] = urhs[1] + alpha * r * muold[1] * u[0]
                urhs[N-1] = urhs[N-1] + alpha * r * muold[N-1] * u[N]
                ##This expression caters for boundary conditions for temperature
                Trhs[1] = Trhs[1] + alpha * r * T[0]
                Trhs[N-1] = Trhs[N-1] + alpha * r * T[N]
                uurhs = urhs[1:N]
                ##Sparse Matrix for Velocity##
                vu = concatenate(([Re+2*alpha*r*muold[i],-alpha*r*muold[i]],zeros(N-3) ))
                Mu = sparse.csc_matrix(linalg.toeplitz(vu,vu))
                unew = linsolve.spsolve(Mu, uurhs)
                u[1:N] = unew
                uold = u
                TTrhs = Trhs[1:N]
                Tnew = linsolve.spsolve(Mt, TTrhs)
                T[1:N] = Tnew
                Told = T
                ##Constants##
                #muold = ones(N+1)
                ##Temperature only##
                #muold = exp(-(Told/(1 + (epsilon*Told))))
                ##Carreau Model only##
                #muold = (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))*ones(N+1)
                ##Viscosity dependence on Temperature and Shear rate##
                muold = exp(-(Told/(1 + (epsilon*Told)))) \
                * (1 + (m-1)*((1 + (beta*sr)**2)**((n-1)/2)))
        return  max(T)
#print Tmax(3.)
def data_plot(n,lbdlist):
        maxiTlist = zeros(len(lbdlist))
        for ilbd in range(len(lbdlist)):

                maxiTlist[ilbd] = Tmax(lbdlist[ilbd],n)
        return (lbdlist,maxiTlist)
Re = 1.0
Pe = 1.0
Br = 1.0
lbd = 0.1
epsilon = 0.01
lbdlist1 = arange(0,3.6,0.1)
lbdlist2 = arange(0,3.5,0.1)
lbdlist3 = arange(0,3.4,0.1)
lbdlist4 = arange(0,3.3,0.1)
tt = arange(0, 4.1, 0.1)
```

```
n1 = -0.5
data1 = data_plot(n1,lbdlist1)
d1 = zeros(len(tt))
for i in range(len(data1[1])):
        d1[i] = data1[1][i]
plot(tt, d1, '--', label = 'n=-0.5',linewidth = 3)
n2 = 0.0
data2 = data_plot(n2,lbdlist2)
d2 = zeros(len(tt))
for i in range(len(data2[1])):
        d2[i] = data2[1][i]
plot(tt, d2, '-', label = 'n=0.0',linewidth = 3)
n3 = 0.5
data3 = data_plot(n3,lbdlist3)
d3 = zeros(len(tt))
for i in range(len(data3[1])):
        d3[i] = data3[1][i]
plot(tt, d3,'-.',label = 'n=0.5',linewidth = 3)
n4 = 1.0
data4 = data_plot(n4,lbdlist4)
d4 = zeros(len(tt))
for i in range(len(data4[1])):
        d4[i] = data4[1][i]
plot(tt, d4,'-+', label = 'n=1.0',linewidth = 3)
legend(loc='best')
xlabel("lambda")
ylabel("Theta_max")
show()
```

Acknowledgements

I am most grateful to God who has been my source of hope, strength, care, provision, protection and his unfailing love throughout my programme at AIMS. I really thank him and praise be to his holy name.

I cannot but heartily appreciate my supervisor Dr. Tirivanhu Chinyoka, I appreciate your corrections, advice and recommendations which enabled me to perfectly complete this essay. I doff my cap to you Sir.

Thanks goes to the visioneer of this wonderful programme Prof. Neil Turok, the director Prof Fritz Hahne and the Logistics and facilities manager Mr Igsaan Kamalie, all the tutors, David Richards, Mihaja and especially Dr Gift Muchatibaya for his fatherly advice may God bless you Sir.

I appreciate all my Nigerian colleagues here at AIMS; a great set of people going somewhere to explode for the glory of God, Word of Salvation Ministries for their ceaseless prayers. I am undoubtedly grateful to my sunshine Awopejo Oluwakemisola, you are appreciated.

Lastly, a river that forgets its source will eventually run dry, this thanks goes to two of the finest people who ever graced this world my parents; Pastor and Mrs E.A Bankole, without them I would not have been this far, my siblings for their constant prayers and encouragement. I am saying thank you, I love you all.

References

[1] Aisoil Techical Service Bulletin. http://www.lubedealer.com/syntheticluberiz/. Thermal Runaway.

[2] T. Chinyoka. Computational Dynamics of a Thermally Decomposable Viscoelastic Lubricant Under Shear. *ASME J. Fluids Engineering*, 130:121201–7, 2008.

[3] F. Durst. *Fluid Mechanics: An Introduction to the Theory Fluid Flows*. Springer Books, 2008.

[4] Churchhouse R. F and Tayler A., editors. *Numerical Solution of Partial Differential Equations: Finite Difference Methods*. Number 0-19-859650-2 in Oxford Applied Mathematics and Computing Science Series. Oxford University Press, Brunel University, 1985.

[5] Myers T. G., Charpin J. P. F., and Tshehla M. S. The Flow of a Variable Viscosity Fluid between Parallel Plates with Shear Heating. *Applied Mathematical Modelling*, 30:799–815, 2006.

[6] Recktenwald W. Gerald. Finite Difference Approximation to the Heat Equation. Available from http://web.cecs.pdx.edu/ gerry/class/ME448/codes/FDheat.pdf , 2004.

[7] Batchelor G.K. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.

[8] Keith E. Gubbins, editor. *Analysis of Transport Phenomena*. Number 0195084942 in Topics in Chemical Engineering. Oxford University Press, Massachusetts Institute of Technology, 1998.

[9] O. K Koriko and A. J Omowaye. Numerical Solution for Frank-Kamenetskii and Activation Energy parameters in Reactive-diffussive- Equation with Variable One-exponential Factor. *J. of Mathematics and Statistics(3)*, 4:233–236, 2007.

[10] Eric Lee, June-Kuo Ming, and Jyh-Ping Hsu. Purely Viscous Flow of a Shear-thinning Fluid between two Rotating Spheres. *Chemical Engineering Science*, 59:417–424, 2004.

[11] O. D. Makinde. Thermal Criticality in Viscous Reactive Flows through Channels with a Sliding Wall: An Exploitation of the Hermite-Padé Approximation Method. *Mathematical and Computer Modelling*, 47:312–317, 2008.

[12] T. M Eldabe Nabil, M. F El-Sabbagh, and M. A. S El-Sayed (Hajjaj). The Stability of Plane Couette Flow of a Power-Law Fluid with Viscous Heating. *J. of Physics of fluids: American Institute of Physics*, 19:094107–1, 2007.

[13] Duarte A. S. R., Miranda A. I. P., and Oliveira P. J. Numerical and Analytical Modelling of Unsteady Viscoelastic Flows: The Start-up and Pulsating test case Problems. *J. Non-Newtonian Fluid Mech*, 154:153–169, 2008.

[14] Izmir Institute Of Technology. Viscosity of Newtonian and non-Newtonian Fluids. Available from Izmir Institute of Technology,Chemical Engineering Department CHE 310 Chemical Engineering Laboratory I, 2009.

[15] D. J Tritton. *Physical Fluid Dynamics*. Van Nostrand Reinhold Company Limited, 1977.